NAVAL POSTGRADUATE SCHOOL

Monterey, California

AD-A247 248

DTIC
ELECTE
MAR 0 6 1992
S B D

THESIS

ALTERNATIVES FOR DEVELOPING
USER DOCUMENTATION
FOR APPLICATIONS SOFTWARE

by

Nancy K. Clark

September, 1991

Thesis Advisor: Norman F. Schneidewind

Approved for public release; distribution is unlimited

92-05421

92 3 03 208

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	

4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b. OFFICE SYMBOL *(If applicable)* 55	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS *(City, State, and ZIP Code)* Monterey, CA 93943-5000		7b ADDRESS *(City, State, and ZIP Code)* Monterey, CA 93943-5000

8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL *(If applicable)*	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c ADDRESS *(City, State, and ZIP Code)*	10 SOURCE OF FUNDING NUMBERS

Program Element No	Project No	Task No	Work Unit Accession Number

11 TITLE *(Include Security Classification)*

Alternatives for Developing User Documentation for Applications Software (U)

12. PERSONAL AUTHOR(S) Nancy K. Clark

13a TYPE OF REPORT Master's Thesis	13b. TIME COVERED From To	14. DATE OF REPORT *(year, month, day)* 1991 September	15. PAGE COUNT 132

16 SUPPLEMENTARY NOTATION

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 COSATI CODES			18. SUBJECT TERMS *(continue on reverse if necessary and identify by block number)*
FIELD	GROUP	SUBGROUP	Computer Documentation, User Documentation, Applications Documentation

19 ABSTRACT *(continue on reverse if necessary and identify by block number)*

The preparation of software documentation is an iterative process that involves research, analysis, design, and testing. The writer must have a solid understanding of the technical aspects of the document being prepared, good writing skills, and, most important, an understanding of the needs of the audience for whom the document is written. This thesis describes the theory and steps taken in developing software user documentation for applications software. The final products are two prototypes of user's guides for applications software programs, specifically WordPerfect 5.1 and dBase IV 1.1, installed on the networks in the Administrative Sciences/Information Systems Computer Laboratories at the Naval Postgraduate School.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS REPORT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a NAME OF RESPONSIBLE INDIVIDUAL Norman F. Schneidewind	22b TELEPHONE *(Include Area code)* (408) 646-2719	22c OFFICE SYMBOL

DD FORM 1473, 84 MAR   83 APR edition may be used until exhausted   SECURITY CLASSIFICATION OF THIS PAGE

All other editions are obsolete   UNCLASSIFIED

Alternatives for Developing
User Documentation
for Applications Software

by

Nancy K. Clark
Lieutenant Commander, United States Navy
B.S., Southern Methodist University

Submitted in partial fulfillment
of the requirements for the degree of
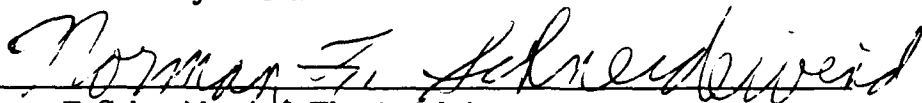
MASTER OF SCIENCE IN COMPUTER SYSTEMS MANAGEMENT
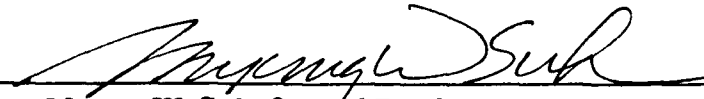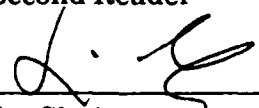
from the

NAVAL POSTGRADUATE SCHOOL
September, 1991

Author: _____
Nancy K. Clark

Approved by: _____
Norman F. Schneidewind, Thesis Advisor

_____
Myung W. Suh, Second Reader

_____
David R. Whipple, Chairman
Department of Administrative Sciences

ABSTRACT

The preparation of software documentation is an iterative process that involves research, analysis, design, and testing. The writer must have a solid understanding of the technical aspects of the document being prepared, good writing skills, and, most important, an understanding of the needs of the audience for whom the document is written.

This thesis describes the theory and steps taken in developing software user documentation for applications software. The final products are two prototypes of user's guides for applications software programs, specifically WordPerfect 5.1 and dBase IV 1.1, installed on the networks in the Administrative Sciences/Information Systems Computer Laboratories at the Naval Postgraduate School.

iii

TABLE OF CONTENTS

I. INTRODUCTION

There is little doubt that the impact of the computer age has affected many in society. We are being flooded with new complex computer systems in which acquiring and retaining information has become increasingly difficult. The technological innovations in the computer field have created problems for both system designers and educators. System designers are faced with the dilemma of trying to create user-friendly interfaces for these new systems, while educators must find ways of teaching complicated information to potential users.

The materialization of these problems has led to much attention being paid recently to the field of user interface design. No longer is it important to just design systems that meet the market demand, but systems must be designed and presented so that it is alluring to the user. As such, the human-factors issue plays an enormous role in the design of new computer systems.

Along with the emergence of new systems, software that takes advantage of advanced technology must also be written and documented. When carefully developed, documentation can be used as a supplemental effort to ease the transition and enhance the relationship between user and machine [Schneiderman, 1986].

Another contributor to the dilemma created by new computer production is the system comprehension issue. As hardware becomes more complex, technical manuals have become increasingly voluminous to accommodate pertinent facts. While enormous amounts of this information become available, more efficient ways of absorbing its content must be developed. Newly developed teaching aids must now encompass the technical sophistication of computer systems in addition to addressing the human-computer interface issues [Bradford, 1983]. Also, training must address a more diverse audience that is not only made up of data processing professionals but also clerical workers and managers. One way to meet these disparate needs is to create user documentation that can be used by the different levels of users and incorporate them into the learning process.

A. PROBLEM BACKGROUND AND DEFINITION

This is the age of the microcomputer. As microcomputers continue to grow in numbers and use, so does the need for communication among them. Spurred by the need to share hardware, software, and data, local area networks (LANs) are expected to proliferate in this environment [Sachs, 1985].

Many offices are now sharing their computer resources through networks, but even in small non-networked office environments, the potential for sharing computer resources is present and the move towards distributed systems is

2

inevitable. This prediction is based on the assumption that most personal computers do not operate in offices alone. In some cases, programs and data in offices are shared among machines and users via the exchanging of disks. Normally as an organization grows, so does its need for more powerful computers, larger storage devices, more memory, increased efficiency in the retrieval and update of data and more sophisticated peripheral devices [Luhn, 1985]. For organizations who want to cut costs, one solution is to share their equipment [Derfler, 1986]. This cost-cutting objective can be achieved through the use of computer networks.

Networks offer an effective, efficient way to share applications and other resources. However, users need to be able to operate the software applications and access shared hardware resources such as printers in this type of environment. To meet these demands, adequate documentation and training must be available.

Network training must be directed toward students, clerical workers, managers, and other users of the computers. These target audiences have varied levels of computer knowledge and experience, and often possess conflicting goals and expectations. Often, user documentation is developed without regard for the audiences' needs. Since not all users have the same learning needs or skill levels, the documentation may be too technical or too simplistic, too specific or too general to be of use. Often, one all-

3

encompassing system documentation is created, intended to meet all the information needs of all the system users. As a result, diverse audiences find the single "system manual" contains the wrong kinds and levels of information [Chinell, 1990].

This thesis analyzes the functional and design issues associated with the development of software user documentation for one of these audiences, the information systems and management students of the Naval Postgraduate School, and produces reference guides for two applications software programs, WordPerfect 5.1 and dBase IV 1.1, as an end product. The study is a report of the developmental process used to design reference guides for applications software used in the Administrative Sciences/Information Systems computer networks at the Naval Postgraduate School.

B. OBJECTIVES

The primary objective of this project was to develop software user documentation for two applications programs resident in the networks (excluding the Apple networks) in the Administrative Sciences and Information Systems computer laboratories which would be used and understood by users of these laboratories.

C. THE RESEARCH QUESTION

What is the most effective method for development of software documentation which will promote maximized use and enable greater productivity of the Administrative Sciences/Information Systems (AS/IS) computer laboratories for Naval Postgraduate management and information systems students?

D. SCOPE

The many problems associated with designing user documentation are the variables upon which this paper focuses. The intent of this study was to investigate the variables needed to produce effective software documentation for users of the AS/IS laboratories. These variables were tested to determine their effectiveness, and modifications then made based on users' needs and recommendations.

1. Audience Description

These reference guides were designed to address an audience that represents an older-than-average college graduate student. Most of these students have been trained in some technical or managerial area in which they have been working for a number of years. Most will be pursuing advanced degrees in the administrative, managerial science, or information systems area, and have limited amounts of free time available because of constraints that course requirements place on them. Computer familiarity varies immensely, with

experience ranging from the novice to the dedicated computer user to those with both job experience and baccalaureate degrees in the computer field.

2. Design Issues

Although knowing the audience you are addressing is of paramount importance, it is also vital to know how much technical information to include, which information to include, and how to organize that information in a meaningful, orderly fashion. The most frequent complaints about computer manuals are:

- Poorly written manuals. Computer manuals are often written by technicians who have no concept of how to present information to users without using technical jargon. The end product is a manual that is inadequate and difficult to understand.

- Important information is hard to find. Computer guides that are not organized around user tasks are often confusing. Users have to expend extra time and effort deciphering the layout scheme.

Considering the audience and issues involved, application documentation developed for NPS users should be brief, task oriented, and written in common English. Two examples of essential user documentation for application software, specifically WordPerfect 5.1 and dBase IV version 1.1, are included as Appendix B and C.

E. LITERATURE REVIEW AND METHODOLOGY

Much has been written to help professionals who write computer documentation to produce better manuals; however, few

address the development of user documentation. These professionals perform a variety of roles in software development settings, such as system designers, system operators, and maintenance personnel, and write different kinds of software documentation, such as design documentation, maintenance documentation, and user documentation. User documentation is the least standardized, supported, and understood of the types of software documentation noted. Automated systems such as CASE tools, rapidly becoming available for producing development and maintenance documentation, have not been developed for user documentation. Development and maintenance documentation writers generally follow well-developed standards for the sequencing, formatting, and content of manuals; not so for user manuals. [Brockmann, 1990]

User documentation is the most difficult for computer professionals to write because it requires communication with people who have widely different backgrounds. It dictates a type of writing that translates computer operations into English that users will understand. [Brockmann, 1990] Program users need documentation as a tool to help them successfully run and understand a program. They want documentation that gives them the instructions, guidance, and reference information they need. [Spear, 1984]

The template approach developed by Dorothy Walsh in 1969, in which the writers merely fill in set templates with

information peculiar to their own system, initially appeared to solve problems of content adequacy and organization. However, it did not have the necessary flexibility of content and organization required to cope with the variability in audiences and purposes. [Brockmann, 1984] Replication of the best procedures used by the best documentation writers, rather than simple replication of document content, offers a method to prepare comprehensive and accurate documentation which addresses and answers the needs of targeted audiences. Using such a structured methodological approach not only aids the developer in organizing the documentation, but is also the primary determinant to producing a well-developed and useful guide.

The procedures used to develop the user documentation for applications programs installed in the AS/IS computer laboratories are based on the Standard Documentation Process (SDP) described by R. John Brockmann. The nine (9) steps involved in the SDP include:

1. Develop document specifications.

2. Prototype the specification.

3. Draft the document.

4. Edit the document.

5. Review the document.

6. Field test the document.

7. Produce and distribute the document.

8. Review the documentation project.

9. Maintain the document.

F. DEFINITIONS AND ABBREVIATIONS

Some of the concepts presented here require working definitions specific to this study. Also, abbreviations and acronyms used in the paper are defined here for the convenience of the reader.

AS/IS - Administrative Sciences/Information Systems

DDP - Documentation Development Process (Williams and Beason)

SDP - Standard Documentation Process (Brockmann)

Software Documentation - Unless otherwise specified, refers to USER documentation rather than design, maintenance, or other types of software documentation.

G. ORGANIZATION OF THIS STUDY

The first chapter is the introductory chapter, including sections which present a general description of the problem, background of the problem, objectives of the research, the research question, the scope, and the assumptions of the research project, a brief description of the research methodology, a list of definitions and abbreviations, and a description of the organization of the study. An overview and a review of research materials and literature relating to the purposes, types, problems, and causes of problems of software documentation, rhetorical orientation of writers, and survey findings on paper documentation is included in chapter two.

Chapter three describes the methods used for executing the research design, comparing documentation development processes, comparing screen and print designs for documentation, and discussing design issues and audience analysis. The fourth chapter discusses the test plan and results and the fifth chapter draws this report together with conclusions and some practical recommendations for developing and maintaining software user documentation, as contained in Appendixes B and C. Appendix A is a matrix of software programs installed on the server computers of the networks in the AS/IS Computer Laboratories. Appendix B is a basic users' guide for WordPerfect 5.1 and Appendix C is a basic users' guide for dBase IV 1.1.

II. LITERATURE REVIEW OF USER DOCUMENTATION DEVELOPMENT

A. OVERVIEW OF SOFTWARE DOCUMENTATION FOR USERS

The field of computer documentation is moving and changing quite rapidly as some of the finest minds in the professions and in academia turn to it as a field of study and research. The industry already has moved beyond merely paper manuals. It is in moving beyond paper that today's writers of paper user manuals will be able to enter the next century with manual-less software in common use. Manual-less software was the objective of the Apple Computer's Macintosh project. Although they didn't fully succeed in being manual-less, the direction in the software industry is to take much of the paper documentation and make it either superfluous because of improved interface design, or put it online using such new organizational devices as hypertext. Manual-less software will become possible as contemporary culture increases its "intuitive" knowledge and sophistication concerning computers, and as the software itself better communicates its purposes and controls to the user. Even now, user documentation writers are not just paper manual writers; rather they are communication specialists who have the necessary expertise to design the communication elements of the "user interface" elements of the software: the messages, the menus, the online

tutorials, as well as the traditional paper manuals.
[Brockmann, 1990]

However, experience has shown that online documentation will not work unless it can be provided within the application program at the point in the user's operation where help is needed. The methods and techniques of communication on paper will not be lost in the transition to a new medium, but rather become more important. In many cases, the qualities of effective online documentation must be abstracted from the qualities of effective paper documentation. For example, the concept that effective online information must allow for multiple access methods of getting to information can be easily abstracted from a book's multiple access methods that range from the "keyword searches" of an index, to a "top-down hierarchical approach" of a table of contents, to a page's headings that allow access to information on a local level. The idea of "aliasing" in keywords or online "links" is nothing more than the application of the concept of using "See" and "See also" in paper book indexes. The principle of effective online documentation is that we have to move away from effective paper documentation, abstract from paper its tricks and techniques, and then reinvent their tricks and techniques in online documentation using different tools.
[Brockmann, 1990]

Several documentation theories will be outlined in this thesis. Often, the best solution to user documentation

problems is an eclectic solution: a little from one writing style, a little from a hypertext-linking philosophy, and so on. Only one methodological guide remains constant and unyielding across all theories: the audience is always right. Thus, as many sides to solutions (and their defects) as possible are presented. [Brockmann, 1990]

More and more documenters are breaking out of the software design organizations in which they merely massage written software design specifications. Now, documenters are getting information from such people as the design team of which they are a member right from the beginning, from actual users in their own environments through the application of documentation specification reviews and early prototype testing, and from fellow documenters in documentation teams and in structured documentation project reviews. Getting more information from people than from books means that negotiating, listening, and getting along with fellow documentation team members, software designers, and users will play much more of a role than ever in the past. [Brockmann, 1990] "Gone are the days when writing was done after a product was complete and writers were given the product specification and told to 'pubs it up!' Today's information developers must work as equal partners with other product developers. The lines between hardware, software, and information are getting blurred with the advent of interactive programming, new input devices, and displayable manuals. For

this reason cooperation and collaboration across disciplines will become even more important and people should start practicing it now." [Grice, 1988]

Computer documentation can be defined as communication designed to ease interactions between computer software and the individuals who operate it. Thus, to write software documentation, you must act as intermediary between the computer software and its users. [Brockmann, 1990]

Inadequate user documentation can greatly increase human errors in computer systems. Robert W. Bailey categorized the major factors for human errors in computer systems. Three categories --environmental

Figure 1 Percentage of human errors directly affected by the documentation.

problems, personnel problems, and organizational accuracy factors-- accounted for 50% of all human errors and are beyond the control of the software or computer designers and documenters. The other 50% are within the control of the designers/documenters, and, of these, 60% are directly affected by the quality of the documentation efforts. Training, written instructions, and the human-computer interface are all affected by the quality of the documentation. [Bailey, 1983]
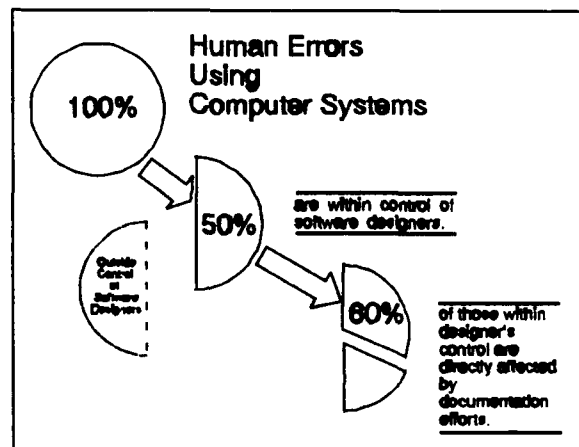
14

During the early years of computers in the 1950s, the standard approach to making computer systems understandable was the users' own trial and error. Early users would have to try one procedure after another, realizing what was the right approach and what was the
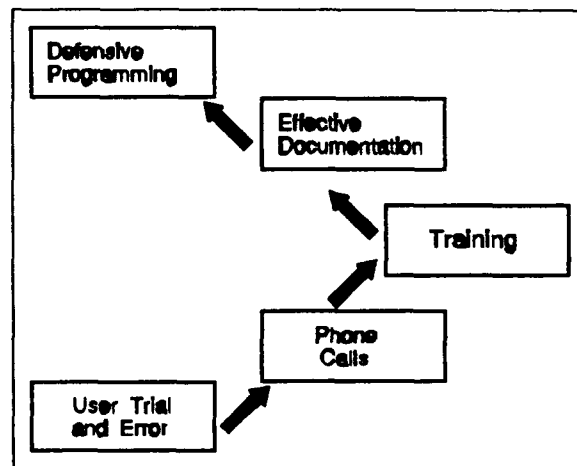

Figure 2 Historical trend in instilling system intelligibility.

wrong one by seeing how many vacuum tubes were blown because of different actions. Intelligibility was instilled in computer systems later by phone call support, training, and effective documentation. The future of computers is represented by "defensive programming," which means anticipating problems and coding to avoid errors before they arise. It includes such techniques as windows-icons-mouses-and-pointers (WIMP). [Brockmann, 1990]

B. RHETORICAL ORIENTATION IN THE WRITING PROCESS

Communication takes place in a context called the rhetorical situation, which includes an encoder (writer) and a decoder (reader or user), each having a purpose for reading or writing. Communication is effective when the message received by the decoder is nearly the same as the message sent

by the encoder. [Pesante, 1991] Rhetorical orientation includes such factors as [Sullivan and Porter, 1990]:

- the writer's model of communication--that is, the writer's beliefs about the way discourse works, the way it ought to be produced, and the way users ought to respond to it.

- the writer's beliefs about priorities in writing--what are important criteria and how do we measure writing effectiveness?

- the writer's attitudes toward authority--where will we look for answers? what authorities should we call upon?

- the writer's conviction to a specific document--to what degree is the writer an advocate for the document? for the system? for the user?

"Noise" that prevents effective communication includes ambiguity, mistaken assumptions, emotional reactions to a topic or word choice, insensitivity of the writer to the needs of the reader/user, overuse of passive voice, long, convoluted sentences, and so on. To be effective, a writer must analyze all the elements of the rhetorical situation and the relationships among them: reader/user(s), writer(s), subject matter, and language. It is especially easy in technical writing to concentrate on the subject matter and neglect the other elements. [Pesante, 1991]

Studies have shown that a writer's use of information is guided by that writer's rhetorical orientation, particularly his/her view of the audience/user. [Sullivan and Porter, 1990] From the perspective of theories of writing and rhetoric, user-centeredness has a solid basis in historical precedent. Rhetoricians have always been concerned with the

importance of occasion in defining the purposes or aims of a piece of communication. Placing the audience at the center of discourse production has also been a significant feature of rhetoric and writing study, so user analysis is a logical extension of this research.
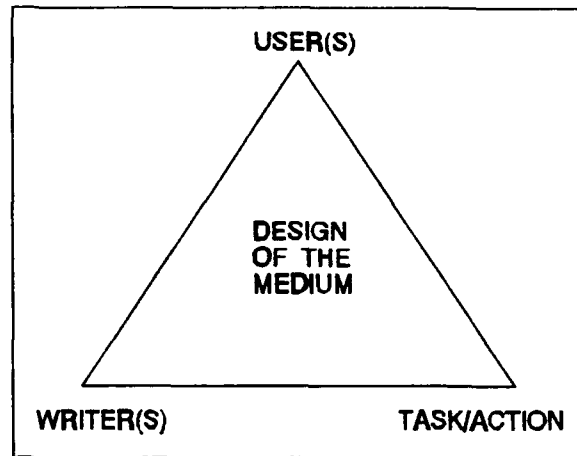

Figure 3. The Communication Triangle of a User-Centered Rhetoric

Modern composition theorists have drawn upon the traditional communication triangle of encoder/decoder/reality to give a conceptual image to the writer/reader/subject triangle. The rhetorical framework of user-centered documentation is reconstructed in Figure 3, to give a clear, conceptual view of the parameters of the discursive territory in question. [Johnson, 1990]

C. SOFTWARE USER DOCUMENTATION PURPOSES

Some of the specific purposes of user documentation are to improve efficiency, to overcome users' fears of equipment or software, and to sell the product.

People need to understand the systems with which they are working. During usability testing of a desktop publishing tutorial, the documentation writer discovered that persistent problems users had with the tutorial were tied to conceptual

issues. His verbal explanations to the users of how the software program works and how it differs from word processing, as well as his description of the systematic operation of the computer system on which the users were working, generated interest in the users and resolved some of their problems. [Sullivan and Porter, 1990]

When users of new software confront a complicated and poorly-organized set of reference manuals as their introduction to a piece of software, they are apt to regret their introduction. On the other hand, if they see a simplified tutorial for the same software, they are more likely to forge ahead. [Brockmann, 1990] The tutorial on desktop publishing mentioned previously was developed as a lock-step, directional guide which maintained a consistent tone, style, and design throughout. Users praised the simplicity and directive approach, particularly early on when they were least confident. Eighty percent of the users reported feeling good about what they had learned about the software program, indicating that they felt confident enough to try the program on their own in the future. [Sullivan and Porter, 1990] Successful software documentation, then, leads to successful first encounters with software and hence to greater acceptance and use. [Brockmann, 1990]

Most people agree that the quality of documentation for the end-user can make the difference between success and failure for a new software product. The manuals are what the

18

customer sees first and therefore they shape the perception of the entire package. [Brockmann, 1990] Also, research indicates that documentation is the most important potential point of difference between software producers. When software products are perceived as fairly difficult to use, documentation can change customers' minds or not. "User friendly documentation shows off user friendly software; together, they win customers and customer loyalty." [Borland, 1984]

D. TYPES OF SOFTWARE USER DOCUMENTATION

User documentation can be classified in two ways: First, it can be classified by content --reference material and tutorial material-- and second, by environment --external documentation and internal documentation. Knowing the different types of documentation will help to make decisions on what to include in a documentation package in response to the audience and the software. [Brockmann, 1990]

Reference material is technical, detailed, comprehensive, and usually organized like an encyclopedia or dictionary for quick retrieval of information. A reference manual should explain what the software can do for the user rather than comprehensively describing the product. An emphasis on product capability in a reference manual allows the user to go beyond the necessarily constrained steps of a tutorial, and combine product features in creative ways. The emphasis of

19

the former is on product capability as opposed to an emphasis on product internal construction, as with the latter. [Brockmann, 1990]

Tutorial material selects from the comprehensive reference material and presents information in a step-by-step fashion. It is usually organized around user tasks or a hierarchy of user needs. [Brockmann, 1990]

A 15-year survey of users carried out by Control Data Corporation, Scientific Data Systems, and Xerox Data Systems resulted in two apparently contradictory findings. Half the users thought manuals had too little detail, and half thought they had too much detail. Two-part manuals, with a tutorial and a reference section, were suggested as a compromise by the survey takers. By clearly segmenting the manual in two parts, the user can choose the coverage of material appropriate for his/her particular situation. [Brockmann, 1990] Borland described much diversity in views of documentation between users with little computer experience and programmers with up to twelve years experience. Borland's "solution manual" was one with a tutorial, a reference, and a "cookbook" (filled with "recipes" to accomplish tasks and procedures for using illustrations, both of the steps and of the result). [Borland, 1984]

Selection of material coverage for a software manual should also be influenced by the "open-endedness" (how much it can be customized, used, and viewed in different ways) of the

software to the user. General database development packages, for example, are open-ended to a greater extent than a spell-checker; thus, the material in the database manual would probably be presented as a reference manual. In a software package which can be viewed or used in only one way, such as the spell-checker software, the more appropriate manual presentation would likely be a tutorial. This is especially true since tutorials tend to limit the users' conceptions of the uses of the software because of the specificity of directions and examples. This may not be desirable for open-ended software packages.

External documentation is meant for audiences outside the corporate or organizational environment in which the documentation is developed. It is usually a more expensive, professional product, being a marketing tool as well as an operations tool. It is usually attractively packaged and filled with graphics. [Brockmann, 1990]

Internal documentation is developed by an organization to be used by people within that same organization. It makes up the bulk of all documentation, yet it frequently fails to receive the necessary time, money, and attention because it is used only inside an organization and is not part of a product to be marketed. Thus, it is not directly related to profit making. Also, internal documentation is not as well designed as external documentation because the writers usually do not seek or receive as much feedback. [Brockmann, 1990] However,

21

an internal manual that is targeted to a specific audience could be superior to a generalized external manual with respect to relevance, simplicity, and brevity.

E. PROBLEMS WITH SOFTWARE USER DOCUMENTATION

There can be many problems with documentation that render it inadequate or ineffective. Some examples of problems are [Brockmann, 1990]:

* Layout and Style Problems: misprints, use of ordinary prose, overuse or underuse of paragraph numbering, lack of or poor highlighting scheme, style that is not conducive to skimming and scanning

* Organizational Problems: not organized to aid the reader's search for information, no preface telling the who, what, and when behind the document, announced order of presentation of material not followed, order of material not intuitive, not apparent, or not supported by graphics, often-used commands not clearly separated, summary of procedures not clearly set out at the beginning

* Audience Analysis Problems: important information missing or unimportant information cluttering explanations, lack of graphics, figures, and other supporting information

* Consistency Problems: programs, commands, functions not having same name throughout, formats and layouts changing, phraseology and wording not staying the same as much as possible, numbering for sections and subsections not consistent, transitions from topic to topic or screen to screen not obvious

* Poor Reference Aid Problems: lack of or incorrect table of contents or indexes, not enough level of detail in table of contents, illustrations, figures, and tables not numbered, titled, or listed

* Update Problems: no plan for updating, handwritten notes used to update

* Language Problems: words such as *files* and *records* used without explanation, no glossary, inappropriate words

used, words used interchangeably which do not mean the same thing, unclear or conflicting instructions

F. CAUSES OF INADEQUATE SOFTWARE USER DOCUMENTATION

Seven factors contribute to the problems causing the production of poor user documentation.

- The change from centralization to decentralization of computer systems

- Institutional limitations

- Inadequate design documentation

- The techniques used in user documentation

- Oversimplifying the writing task in many how-to books and professional journals

- Fighting against rather than harnessing the learning behaviors adults spontaneously adopt

- Natural egoism

Many writers have had difficulty adjusting to the change in the place and function of user documentation. In the 1950s through the early 1980s, computer systems were mostly centralized and surrounded by software specialists who could translate any user documentation that users did not understand. With the shift toward decentralization, where microcomputers and workstations stand alone throughout organizations and geographic locales, software specialists are not available to translate or train at each node. Thus, user documentation must also be able to stand alone. Too many writers continue writing in the centralized frame of mind. [Brockmann, 1990]

Even when documenters are alert to the needs of documentation in a decentralized environment, institutional factors may still prevent them from producing effective computer documentation. First, training and education are lacking in many organizations. Second, company standards which support and enhance good, effective documentation often do not exist. Similarly, if examples of documentation formally presented in standard development methodologies or informally circulated in an office are not examples of good documentation, training and official company standards may go for naught. Finally, good, effective technological support of the documentation effort enhances the likelihood of good documentation. The various iterations of a document should be completed as faultlessly and as quickly as possible. Speed of production is crucial because good documentation is the result of continuous rewriting. The longer and more laborious the process, the less inclined documenters will be to redraft and rewrite. Hence, the more powerful the tools that are put in the hands of the documentation developers, the better the final document will be. Underlying these institutional factors are management support and encouragement. Good or bad documentation and the climate producing either are largely a function of management. [Brockmann, 1990]

To write an effective piece of computer documentation, a writer needs full and complete information on the design of the system or the program. Without a solid foundation of

complete, clear, and accurately-written design documentation, user documentation writers must expend more time and effort in interviews with designers, making educated guesses, and repeatedly reviewing and rewriting. [Brockmann, 1990]

The techniques required for writing user documentation differ radically from those required by essays or the like. The basic difference is that between arranging information for sequential access and arranging it for random access. Computer documentation must be able to be easily scanned and skimmed. [Brockmann, 1990]

Oversimplification occurs when documentation writers turn for help to commercially published instructional texts which do not adequately alert them to the effects that audience variations have on documentation projects and products. As an example, the use of templates, in which writers needed only to fill in set templates with information peculiar to their own system, initially appeared to solve problems of content adequacy and of organization. As template use developed, however, it essentially confused sophisticated data processing users who had extensive prior knowledge with users who had no such knowledge and needed much more. [Brockmann, 1984] Today, new problems may be caused with the advent of industry-wide, corporate-endorsed, research-based guidelines and user interface standards. These standards may cause problems in four ways [Brockmann, 1990]:

- First, to be generally applicable, they often fail to be sufficiently specific to the users' tasks

- Second, they can foster a superficial consistency

- Third, research data are currently too incomplete to support all the rules in these standards and so "best guesses" are packaged indistinguishably with "soundly researched principles"

- Fourth, and most important, guidelines, rules, and standards invite documenters to forgo testing with real users.

Some adults resist explicitly addressing themselves to new learning. Two paradoxes, motivation and assimilation, are described as explanation for this kind of behavior. The motivation paradox is the "production bias" people bring to the task of learning and using computers. The chief goal is throughput, reducing motivation to spend any time just learning about the system; so, when situations occur that could be more effectively handled by new procedures, people are likely to stick with those they already know regardless of their efficacy. The assimilation paradox is that people apply what they already know to interpret new situations. Though often helpful, irrelevant and misleading similarities between the new and old information can blind learners to what they are actually seeing and doing, leading them to draw erroneous conclusions or preventing them from recognizing possibilities for new functions. [Carroll and Rosson, 1987] In the design of documentation that takes the "systems" approach, which focuses on step-by-step procedures in which the reader is

26

expected to be passive and just follow along, these two paradoxes can become problematic. [Brockmann, 1990]

Natural egoism is the final factor that can adversely affect documentation. A writer will not be effective until he/she is able to empathize with the readers and recognize that the readers approach software documentation with different backgrounds, expectations, training, and education. [Brockmann, 1990]

G. SURVEY FINDINGS REGARDING PAPER DOCUMENTATION

Table 1 summarizes the common recurring findings of the four marketing surveys described in this section.

Major Consistent Findings for Paper Documentation	Xerox, Control Data, Scientific (Maynard, 1982)	AT&T (1986)	Microsoft (Borland, 1984)	PC-User Group (Wilton, 1985)
More task-orientation	X	X	X	
More tutorials		X	X	X
Improved reference aids	X	X	X	X
More accuracy		X		X
More illustrations		X	X	X

Table 1. Recurring findings in user surveys of paper documentation

The 15-year survey of users carried out by Xerox, Control Data, and Scientific Data Systems [Maynard, 1982] and a

parallel survey carried out with military users [Beard and Callamars, 1983] showed the major complaints to be:

- Manuals were software-oriented rather than function-oriented

- Manuals did not have enough examples

- Manuals did not have enough reference aids

An independent market research firm conducted an external documentation market survey for AT&T in 1986. The survey identified features of documentation that both users and dealers thought were important factors in selecting one software package over another. Major themes were that information in AT&T manuals should be:

- Easy to find: better reference aids were recommended

- Easy to understand: not assume too much, have graphics, and be task oriented

- Complete, accurate, and current

- Indexed: absence of an index was a definite reason to avoid purchasing a software product

Microsoft Corporation conducted a documentation survey in 1984 [Borland, 1984] which found that end users wanted:

- task-oriented tutorials.

- screen illustrations and terms explained in glossaries.

- reference cards which listed first all the commands and then the tasks with commands used to complete them.

- a feature-oriented/command index as well as a task-oriented index.

- a task-oriented organization.

- a reference manual that comprehensively described all the features of the product.

• trouble-shooting guides.

A PC User Group conducted a survey of 241 microcomputer owners [Wilton, 1985]. Table 2 portrays the contrast between what readers said they wanted and what was actually delivered.

Do you agree or disagree?	Agree	Dis-agree
Manuals should accommodate all users vs. Manuals do accommodate all users	86 16	7 58
Tutorials are usually helpful vs. Many manuals omit tutorials	66 65	12 8
Illustrations should substitute more for text vs. Illustrations are adequate in number	70 18	15 54
Information is easy to find vs. Manuals bury important information	11 89	67 4

Table 2. Contrast between what users want and what users get.

29

III. FUNCTIONAL ANALYSIS AND DESIGN

A. COMPARISON OF USER DOCUMENTATION DEVELOPMENT PROCESSES

The software user documentation development process is defined and executed differently, depending on where you are and who you talk to. There seem to be as many processes as there are organizations who develop software or authors who write about developing software. Three treatments of software user documentation development processes are presented by Brockmann [1990], Weiss [1985], and Williams and Beason [1991].

Brockmann [1990] revised his original seven-step Standard Documentation Process (SDP) which came out in 1986 to accommodate the many developments in the area of user documentation. His nine-step SDP version 2.0 now gives information on CASE tools, new programming technologies, research on layout, format, typography, and use of color, desktop and electronic publishing, new documentation databases, and the effects of new techniques, technologies, and ideas, such as team writing styles, document prototyping, minimalist design philosophy, hypertext, and mass storage devices like CD-ROM and magneto-optical storage. The SDP replicates many tried and tested procedures used by well-respected, successful documentation writers. Figure 4 and the

following list give an overview of what is involved in the
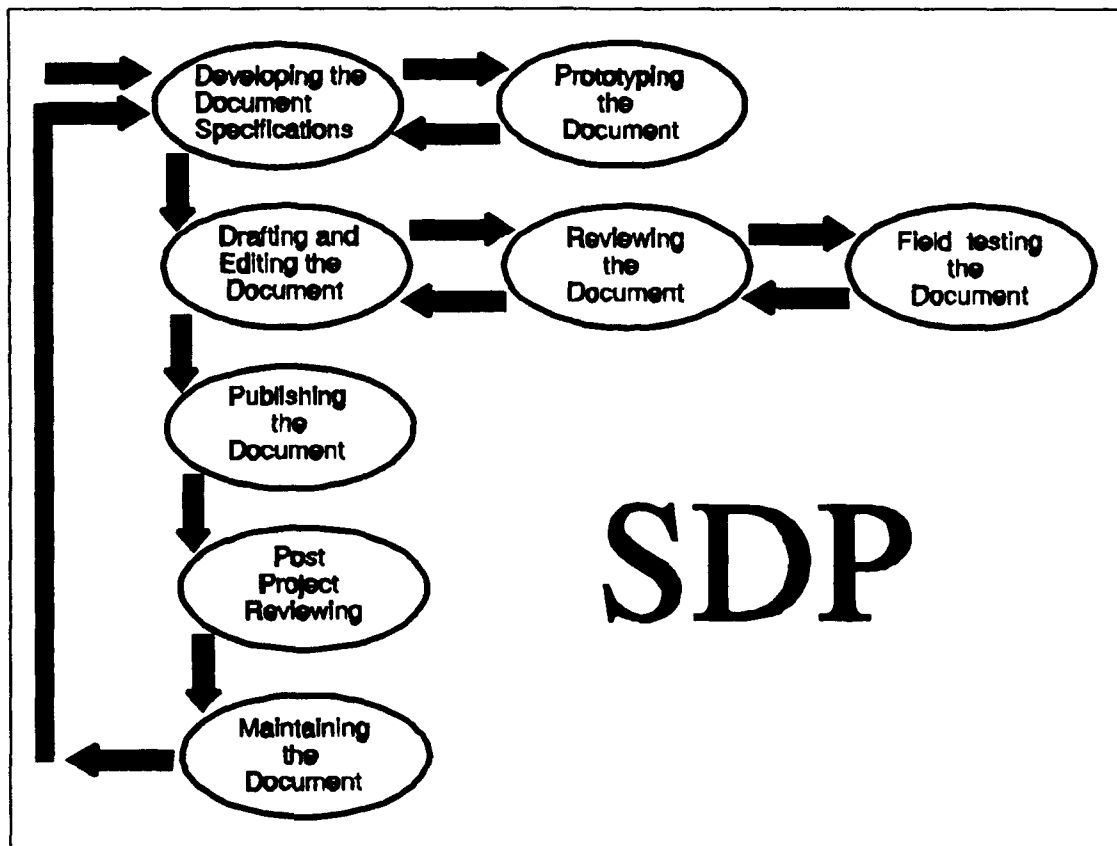
SDP. [Brockmann, 1990]

Figure 4. The Standard Documentation Process (Version 2.0) [Brockmann, 1990]

- Step 1. Develop Documentation Specifications. In this first step, planning the documentation occurs in two passes. The first pass is the development of a Library Specification that contains a brief description of all the documents involved with a particular software program or system. This plan gives an opportunity to communicate the "big picture" of the whole writing project to management or clients. The second pass is the development of the Individual Document Specification. This second blueprint follows the Library Specification and communicates the specific plans for a single document to management, clients, and users. Eleven activities are involved in the creation of these blueprints: breaking down the documentation in the library by tasks, using minimalist design principles, planning for an audience, analyzing the purpose of the documentation, organizing the material, developing a product visualization, picking the

31

appropriate media, deciding on page format and layout, planning for updating, considering the competition, and estimating cost and time requirements.

- Step 2. Prototype the Specification. Since the ultimate test for paper or online documentation is usability, document designers should iteratively test their pages and screens with users. Prototyping is repeatedly done with a document as it is being developed, but it basically has four steps: prepare for the test, instruct the testers, run the test, and analyze and apply the results. The results of prototyping should give guidelines for depth of coverage, vocabulary, readability, and organization.

- Step 3. Draft the Document. Once the specifications have been created, approved, and tested, it is time to draft the document. Seven activities make up this step: overcoming internal and external writing blocks, using a writing style that is designed to match adult reading behaviors, using reader-based writing techniques, developing effective graphics, creating reference aids, developing the documentation packaging, and planning for updates.

- Step 4. Edit the Document. Now that the document is drafted, it is revised so that it effectively and efficiently gets its message across. This is primarily accomplished by using levels-of-edit techniques.

- Step 5. Review the Document. Once the document is drafted and edited, it is sent out for review. To have an effective review, carefully choose reviewers and the time to review, show reviewers how to review, and give them feedback.

- Step 6. Field Test the Document. A part of every document's review should be a field test of a draft of the whole document. Where prototyping examined the pieces of a manual or online document during their creation and assembly, field testing examines how the document works as a whole. Accessibility, navigational problems, and consistency are primary areas of concern here. In conducting a field test, carefully choose field testers and the time to field test, run both an in-house "controlled" field test and an external field test, and provide feedback to field testers.

- Step 7. Produce and Distribute the Document. Once the document is drafted, revised, and reviewed, it is produced in a form suitable for distribution. With paper and online publishing mechanisms ranging so widely, and

multimedia publishing becoming more and more prevalent, preparing text via an SGML-like tagging system becomes essential.

- Step 8. *Review the Documentation Project.* Once a document is complete, and before beginning a new project, analyze what went right and what went wrong during the process of developing the documentation so that improvements to the process can be implemented during the next project, and mistakes and problems thereby alleviated.

- Step 9. Maintain the Document. Even when the document is distributed, the task is still not completed because the document must be updated. To do this effectively, responsibility for updating a document should be clearly assigned. Distribution of the document should be tracked so that one knows where to send updates and the changes in the updates should be clearly indicated.

Weiss [1985] describes five phases of user documentation in his Structured Process, all of which he believes necessary for effective, usable documentation. By structured, he refers to a formal, top-down decomposition of the user documentation development process into a development model which is designed so the components of the process are interconnected in the "best possible way." [Weiss, 1985] Figure 5 is a data flow diagram of the five phases, showing the development flow for user documentation. A brief description of the five phases of the Structured Process follows.

- Phase 1. Analysis. Define just what manuals and other information products the users and operators need. The earlier in the life of the system the analysis takes place, the better. Ideally, the documentation analysis, which often culminates in a Publications Plan in large projects, should occur as part of the original system development plan, but it is never too late to analyze the remaining documentation need.

- Phase 2. Design. Prepare detailed outlines of each manual or other information product. This phase starts
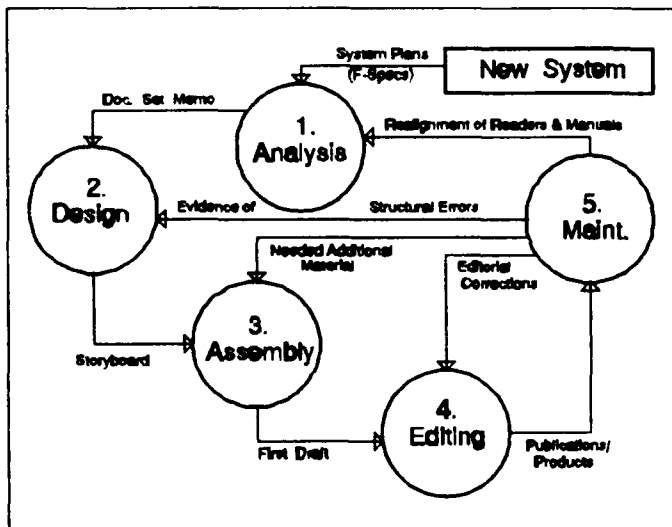
Figure 5. Data Flow Diagram for Developing User Documentation using A Structured Process. [Weiss, 1985]

- with the preparing of conventional outlines, but then proceeds to the creation of "structured outlines" and "storyboards" -- working models of the documents that can be tested and revised before the first draft is written. The most difficult structural and organizational problems must be corrected before the first draft is written.

- Phase 3. Assembly. Convert the storyboard to a work plan and write the first draft. In the structured approach to documentation, writing the first draft is a little like writing the code in a structured program: that is, the writers do little more than fill in missing details, according to a strictly-followed plan, the "storyboard."

- Phase 4. Editing. Test the first draft for clarity, correctness, and readability. In this approach, questions of language and style are more than matters of esthetics; rather, the purpose of this phase is to apply principles of editing that make the manual easier to use, and therefore less likely to cause a "failure" (defined as what occurs when an operator or reader us unable to work the system because of a bug in the manual). In many cases, this phase culminates in a test with "live" readers.

- Phase 5. Maintenance. Track what needs to be changed in the information products and, when appropriate, make the changes. Because all manuals are flawed or out-of-date

34

(without exception, according to Weiss), the last phase of documentation is to monitor what should be added, removed, replaced, or repaired. The craft of maintaining user documents is knowing what changes must be made and being able to distribute and incorporate those changes in a manner that does not generate confusion and additional errors.

Williams and Beason [1991] view software development projects as having nine phases. These nine phases somewhat

Table 3. User Documentation in the Software Development Process. [Williams and Beason, 1991]

Phase	User Documentation
1 Feasibility Analysis	
2 Requirements Definition	
3 Alternatives Analysis	
4 Functional Specifications	
5 Preliminary Design	Documentation plan
6 Detailed Design and Construction	Outline(s) for user documentation First draft(s) of users' manual(s) First draft(s) of online help system
7 Verification (alpha and beta tests)	Reviews of first drafts Usability tests of tutorials and procedures guides Review, linking, and testing of online systems Final draft(s) of users' manual(s) First and final drafts of quick reference pieces
8 Implementation	User documentation completed
9 Maintenance	Revisions and addenda

correlate with the seven phases of the Standard Software Development process from the IEEE Software Engineering

35

Standards reference [1990]: Concept, Requirements, Design, Implementation, Testing, Installation and Checkout, and Operation and Maintenance. According to Williams and Beason, the user documentation development process begins during phase five of a software development project and continues throughout the remainder of the life cycle of the project. Table 3 shows how Williams and Beason [1991] believe user documentation fits into the overall software development process. As you can see, their Documentation Development Process generally comprises five phases, as follows:

- Phase I. Documentation Planning. Locate and review existing information and confer with team members. Decide how many and what types of individual documents (manuals or other printed pieces) and online materials are needed. Decide on the goals of the documents. Write a profile of the audience. Determine production methods, including the means for creating illustrations, producing a final draft, and reproducing or printing the required number of copies. Describe the physical appearance of the document, put the plan on paper, and get it approved. Draft a schedule and get it approved. Create a style guide.

- Phase II. Outlining and First Drafts. Review information in the documentation plan about readers and their needs, and the goals of the document. Decide how to organize the document. Draft a preliminary or working outline of the printed documents, including quick reference materials. Draft a preliminary outline of the online materials. Review the outlines and revise them if necessary. Get the outlines approved. Write the first draft. Write or review the first draft of the online materials. Make a preliminary list of illustrations. Read and revise the first draft. Update the list of illustrations. Send the draft out for review.

- Phase III. Subsequent Drafts, Usability Testing, Final Drafts. Incorporate comments and corrections from the review. Do any necessary rewriting. Make copies of completed illustrations and insert them in the draft. Proofread and correct the draft, covering both text and illustrations. Send the draft out for review. Review and

correct online materials. Conduct usability tests of tutorials and procedures guides. Review, link and test online systems. Incorporate comments and corrections from the prior review of drafts. Read all text and review illustrations, checking for flow, clarity, and completeness. Write quick reference materials. Get final okays on changes from reviewers. Have quick reference materials reviewed and proofread. Check text and illustrations for consistency, and proofread for typographical, spelling, or placement errors. Indicate spaces for illustrations. If necessary, mark headings, words, or phrases that need special emphasis, and page breaks. Review online materials for the final time. Correct quick reference materials.

- Phase IV. Production. Produce the text by typewriter, word processing software, or computerized typesetting. Proofread the text. Make up pages, merging text and graphics. Check for continuity and positioning of illustrations. Number the pages of the document if needed. Prepare the table of contents and index. Proofread page numbers for the index and table of contents. If the document is being professionally printed, check the blue line (sample of printed document) for accuracy, consistency, and placement of text on pages. Print or duplicate the required number of copies. Assemble, bind, and distribute documents.

- Phase V. Maintenance. Prepare, incorporate, and distribute revisions and addenda on an ongoing basis as necessary.

Although the number and definitions of the phases and steps in each of these processes vary, all have commonalities which must be viewed as mandatory in any software user documentation development process. A comparison of the three approaches to the documentation process is provided in Table 4. However the steps or phases are organized, the actual processes all contain the elements of planning, designing, drafting, rewriting, testing, producing, distributing, and maintaining the software user documentation developed.

Table 4. Comparison of Software User Documentation Development Processes.

SDP [Brockmann, 1990]	Structured Process [Weiss, 1985]	Document Development Process [Williams & Beason, 1991]
1. Doc Specs	I. Analysis	I. Documentation plan
2. Prototype	II. Design	
3. Draft Doc	III. Assembly	II. Outline & First Draft(s)
4. Edit	IV. Editing	III. Reviews, Tests, Final Drafts
5. Review		
6. Field test		
7. Produce/Distro		IV. Production
8. Review Project	V. Maintenance	V. Maintenance
9. Maintain		

Since the development methods studied displayed commonality, the decision on which method to use for development of the user's guides for the AS/IS Computer Labs at NPS was based on versatility, applicability, complexity, completeness, and currency of the method. This researcher was looking for a method which has the following attributes: 1) versatile enough that it could apply across most development situations; 2) easy to understand; 3) speedy assimilation; 4) sufficiently complete; and 5) applicable to military graduate students. Also, this researcher desired a method which had been developed or updated within the last couple of years so that its techniques would incorporate industry advancements and evolutions. For these reasons, a combination of the SDP formulated by R. John Brockmann [1990] and the Documentation

Development Process formulated by Williams and Beason [1991] were selected as the guides for creating the user's guides contained in the appendixes of this paper.

B. COMPARISON OF SCREEN AND CATEGORIES OF PRINT DESIGN

The content-classified types of software documentation (reference and tutorial) discussed in Chapter 2 can be further categorized into five basic classifications [Williams and Beason, 1991]:

- Tutorial: Teaches basic program functions through controlled "lock-step" practice sessions

- Procedures guide: Explains and gives step-by-step instructions about how to perform all the functions of the program

- Reference material: Describes in detail commands, functions, fields, key assignments, and/or messages

- Quick reference piece: Lists the most frequently used commands, functions, or key assignments (may be a card, keyboard template, small guide booklet, or single-page document)

- Online help system: Displays information on the screen while the program is running

The following chart aids selection decisions with regard to the categories of documentation to produce [Williams and Beason, 1991]:

TYPES	USE WHEN	ADVANTAGES
Tutorial	Users are novices	Builds confidence
	Users must teach themselves	Lets users practice
	Users need to get started quickly	Allows quick, user-friendly use of program features
	Program is complex or interface is intimidating	
Procedures guide	Users have some experience or program is simple	Allows users to choose only procedures they need
		Information is complete, arranged in task-oriented groupings
Reference material	Users know how to use features and are familiar with the interface	Allows quick access to details
		Allows users to approach information from many angles
Quick reference piece	Users are experienced with the program	Quickly reminds users which commands, functions, or keys to use
Online help system	Users need information while running the program	Allows users to get assistance without looking away from the screen

The decision about which categories of documentation to produce depends on the needs of the audience and on the budget and schedule. For documenting only an application, a procedures guide may be sufficient; for documenting a programming language, a reference manual may be adequate. Generally speaking, no one category of documentation can please all types of users. [Williams and Beason, 1991] For the purposes of this study, task-oriented procedures guides, also called a user's guides, were selected for production, based on the audience analysis and limited budget and schedule, and because effective, complete online documentation was already available via the online HELP features offered by both WordPerfect 5.1 and dBase IV 1.1.

C. DESIGN ISSUES

Normally, the first exposure a new user will have to a system is through user's manuals. Careful planning must be devoted to the design of documentation since inadequate design results in end-user dissatisfaction. Readers only use documentation to get their job completed when it requires some kind of computer assistance. Thus, the best design for software documentation is the one that fits the users' methods of working and requires the least attention and learning.

In designing the user's guides for the AS/IS Computer Labs, the researcher focused on organization, content, layout, and language. It is thought that these elements are the four

41

factors that most impact the effectiveness of the documentation design. [Gleason, 1986]

1. Organization

Widely used in the industry until recently, a software-internals orientation approach to documentation design concentrated on the structure and facilities of a program rather than on the user's use of that program. This approach received extensive criticism in the 15-year survey of users carried by Xerox, Scientific Data, and Control Data corporations. Users almost always preferred task-oriented manuals because a software-internals orientation forces users to center their business duties around the software rather than vice versa and because users must know the structure of the software before being able to use the documentation. [Maynard, 1982] On the other hand, a task orientation is based on an analysis of the user's use of the program and is limited to what information is required to do a specific task using the program. Thus, the focus is turned from the system to the users using the system in their daily work. [Brockmann, 1990] "Having the user at the center...allows for concentrated efforts from diverse fields toward a common goal: the development of usable documents for whatever medium may come our way." [Johnson, 1990]

The IEEE Standard for Software User Documentation [IEEE Std 1063-1987, 1988] states that:

"Users of software need documents either to learn about the software (instructional mode) or to refresh their memory about it (reference mode). Instructional mode documents may be either information- or task-oriented. Information-oriented documents give the reader information needed to understand the computer software and its functions; task-oriented documents show the reader how to complete a task or reach a goal."

The end products of this paper, two applications user's guides, are more for reference than instruction, but contain some general information about the application software and its functions. The primary purpose of each was to guide certain NPS students on their own or, if desired, as a supplement to instructional courses, through the basic procedures of word processing (in the case of WordPerfect 5.1) and working with a database management system (in the case of dBase IV 1.1). Thus, a task-oriented approach was adopted.

A task orientation, in which the designer selects content and employs an organization appropriate to a user's work needs, points to use of the minimalist design philosophy. The goal of a minimalist design philosophy is to present material appropriately to the actual ways adult learners learn rather than fighting against their natural tendencies the way a "systems" (e.g., software-internals) design philosophy does. [Brockmann, 1990] Research has verified that adult learners:

- Are impatient learners and want to get started quickly on something productive

- Skip around in manuals and online documents and rarely read them fully

- Make mistakes but learn most often from correcting such mistakes

- Are best motivated by self-initiated exploration

- Are discouraged, not empowered, by large manuals with each task decomposed into "irritating" subtask detail

The key idea is to present the smallest possible obstacle to learners' efforts by providing less overt training structure. [Carroll, 1990] The minimalist design tips used in creating the user's guides in Appendix B and C include [Brockmann, 1990]:

- Minimize secondary features of manuals and online documents (overviews, introductions, summaries, etc.).

- Focus on what readers need to know to immediately apply it to productive work.

- Make it easy for the reader of a page to coordinate the documentation with the screen information by grouping instructions and cursor movement/navigation key tables together by the screen to which they apply

- Use what the readers already know by continuously linking new information to it

The DDP exhorts designers to organize the guide and group procedures to reflect the way users will use the program, such as listing the procedures for entering information before listing the procedures for editing that information. Additionally, designers should include all the information needed to successfully complete each procedure. Additionally, each procedure module should be organized internally, so the modules will be consistent. [Williams and Beason, 1991] In the user's guides appended to this paper, for example, each procedure module contains a heading (name of

44

procedure), brief introductory paragraph, and numbered steps of the procedure.

Brockmann [1990] lists two primary principles of organization, both of which were followed in organizing the text material in the appended user's guides.

- Make the organization of material apparent to readers. Essentially, that means that with words, graphics, or layout, "Tell the folks what you're going to tell 'em before you tell 'em."

- Organize documentation in ways expected by the readers. Using general-to-specific and explanation-to-specific conditions works effectively; that is, instructions state a general procedure which is applied to the specific context. Also, readers expect information to be presented in chronological order, most-important-to-least-important order, order of need, and order of difficultly.

The length of a user's guide varies depending on the subject matter. This researcher tried to limit the WordPerfect guide to fifteen pages of material, which would adequately cover all basic functions to be performed by the target audience. Since a database management software program is generally so much more complex than a word processing program, the dBase IV guide required approximately thirty pages to adequately cover the material needed by the target users.

2. Content

The content is the part of the manual that describes operations. It focuses on commonly-used tasks and its productivity is measured in terms of relevancy to the user. The key to selection of tasks to be covered was the "80-20

Rule." The 80-20 Rule is that the user's guide should cover the 20% of the tasks that are used 80% of the time.

An overview of the project also helps define what information should be included. Elements that should be considered are [Williams and Beason, 1991]:

- General purpose of the project

- Intended users of the software

- Features of the software

- Features that are outstanding or that make it different from other similar software

- Operating system and other related software

- Computer and other related hardware

- Network or larger system the software may be part of

Only the content that the user needs should be included. The manual should be as brief as possible, but not at the sacrifice of pertinent information.

3. Layout

A good layout can make a manual more readable and give the writer the means for presenting information clearly and cleanly. A layout for a software manual must meet two goals: to make it easy for users to absorb information on the first reading and easy for them to locate specific bits of information later when they may need them. Generally, to design a manual which meets both these requirements, the writer must [Williams and Beason, 1991]:

46

- Format the separate elements consistently so the readers can identify them at a glance. All procedures should look alike.

- Use a generous amount of white space (blank space) on the page. Information stands out on a page with white space and the pages are less tiring to read.

- Use headings to show the structure and hierarchy of the information. Format them so they stand out from the text and so the subordination is clear.

- Make columns of text no wider than 65 to 70 characters per line (4-5 inches). Wider lines of type are hard to read.

Documents are usually made up of three major sections: the front matter, the main text, and the back matter. The front matter includes all or some of the following: title page, copyright or acknowledgements page, table of contents, list of figures and/or tables, symbols and conventions page, and installation and start-up guidelines. The main text is the introduction and main body of the document. The back matter contains any appendixes needed, a glossary, and an index. [Williams and Beason, 1991] For simple, in-house documents, such as those created for this study, only the parts considered absolutely essential are included.

4. Language

If you want users to get the most out of the program, use language that is clear, strong, and direct. Williams and Beason [1991] and Brockmann [1991] provide many guidelines which were considered when formulating the user's guides:

- Use nonsexist language: Use generic titles and descriptions (e.g., chair instead of chairman) and nonsexist pronouns and adjectives. Some techniques to help include addressing readers directly (e.g., You can...

instead of He can...); use the plural form rather than the singular (e.g., programmers keep their disks... instead of a programmer keeps his disk...); if the title must be singular, substitute an article (a, an, the) for a pronoun (e.g., the operator enters a password rather than the operator enters his password); repeat the title of a person rather than using a pronoun; if there's no other way, use he/she or him/her.

- Use plain language: Use short sentences predominantly and plain one- and two-syllable words whenever possible.

- Eliminate unnecessary words: Avoid noun clusters (use classroom instead of structured learning environment); avoid prepositional phrases (use because instead of as a result of); avoid redundancies (use repeat instead of repeat again); avoid wordy phrases (use truth instead of plain, unvarnished truth).

- Use active verbs: Active verbs make your style forceful and direct; passive verbs weaken your language and makes it seem vague and lacking in authority.

- Choose the proper tense: Use the present tense most of the time; it's easy to try to use the future tense, which weakens your writing (e.g., To write a program... instead of If you're going to write a program...).

D. AUDIENCE ANALYSIS

The IEEE Standard for Software User Documentation [IEEE Std 1063-1987, 1988] prescribes that a software user document shall be keyed to its audience because the identified audience dictates the document presentation style and level of detail. The intended audience is to be identified and the different ways the users interact with software are to be considered when preparing user documents.

Williams and Beason [1991] offered an audience profile description list which was useful in analyzing the anticipated

48

users of the user's guides being prepared for this study. The elements included were:

- Level of computer expertise: The range for this audience varied immensely, with experience extending from the novice to the dedicated computer user to those with both job experience and baccalaureate degrees in the computer field. Few are expected to have had experience on a network.

- Occupation: Students in the AS Department at the Naval Postgraduate School who are pursuing advanced degrees in the administrative, managerial science, or information systems area. Most have been trained in some technical or managerial field in which they have been working for a number of years.

- Knowledge about the field and subject: Although many have at least an idea of what word processing is about and have even used such programs previously, most potential users of the guide are presumed to have little experience with WordPerfect 5.1. Most potential users have not had any experience with a database management system or, specifically, dBase IV 1.1.

- Position in organization or field: The prospective audience is students and faculty at an academic institution (NPS).

- Level of education: The prospective users have at least a baccalaureate degree.

- Age group: The students who will use the guides are older than average for graduate students, but a variation in age from mid-twenties to mid-forties is expected.

- Reasons for using the program: Students are expected to use the WordPerfect program to create reports and research papers; in some cases, to complete an introductory course in WordPerfect 5.1; and to use the program in a variety of courses. They are expected to use dBase IV for introductory courses in database management and dBase IV.

As the students progress in their curriculums and become more familiar with the applications programs, especially WordPerfect, they will "grow" from novice (or wherever they

started on the scale of experience with the programs) to intermediate users who then become experienced users and so on. Thus, what pleases them in the beginning may not always please them six months down the road. Also, as they progress, the more experienced users are graduating and leaving, and new students are arriving. Thus, a varied audience is guaranteed. No single user's guide will be able to fill every student's needs.

IV. THE SOFTWARE USER DOCUMENTATION TEST PLAN

A. TEST PURPOSE

The best type of review of a user manual comes from the users themselves. This step is referred to as field testing by Brockmann [1990] and usability testing by Williams and Beason [1991]. In field or usability testing, the users of the documentation try using a document to see if it is effective and can stand alone. Testing helps provide information on how to improve the document before it is completed. Changes can then be made and the guide retested before the completed version is released. This has long been a standard method of testing computer systems and programs.

B. TEST OBJECTIVES

The main objective is to identify problem areas in the manual while it is still in the development stage. Usability testing is designed to help find problems in the manual's wording, flow, and layout. It should indicate whether the writing style used in the manual can be understood by the intended audience, help identify steps that may have been inadvertently left out, and point out descriptions that do not match tasks. The testing can also provide information from test subjects on what areas they would like to see covered in the guide.

51

C. TEST PLAN

This researcher followed six steps in planning the usability test [Williams and Beason, 1991] and [Brockmann, 1990]:

- Step 1. Find volunteers who match your audience profile. If this is not possible, simulation of a typical audience is acceptable.

- Step 2. Write an instruction sheet, listing simple tasks for the testing subjects to perform; the tasks should use basic and representative functions of the software.

- Step 3. Decide on a reasonable length of time to give the volunteers to complete each exercise.

- Step 4. Choose observers and brief them about your test purpose, their role, and so on.

- Step 5. Make arrangements for the use of work areas for the day of the usability test. Make sure they'll be properly equipped, not only with the relevant computer and software but also with adequate lighting and desktop or table space.

- Step 6. Prepare copies of the documentation that include a table of contents and an index.

After the test, results should be compared among the test administrator and observers to see which areas consistently caused confusion among the volunteers and which areas caused the greatest degree of frustration.

D. TEST ADMINISTRATION AND PROCEDURES

The tests were carried out using the preceding plan, as follows:

- Step 1. A total of seven NPS students were used to test the documents. They were tested individually or in a group of three.

- Step 2. For the WordPerfect guide, a short letter was invented for the testing subjects to type in a document, save, retrieve, and print. For the dBase IV guide, a short database of names, addresses, and phone numbers was provided so the subjects could build a database structure, index it by last name, query it, and generate a quick report.

- Step 3. The WordPerfect test was expected to take approximately one hour and the dBase IV test twice that. In fact, the WordPerfect test required one hour and ten minutes while the dBase IV test was completed in one hour and forty-five minutes.

- Step 4. One observer was used. This observer was an NPS student who served as a network laboratory assistant for the AS/IS Computer Labs and who also wrote a software user document for one of the applications installed on the AS/IS networks.

- Step 5. The tests were conducted in one of the AS/IS Computer Labs, on the Token Ring network in I-224. All resources needed for the testing (computers, printers, programs, desks, etc.) were already set up. The tests were conducted on two consecutive weekends when the labs were available and mostly empty.

- Step 6. Sufficient copies of the user's guides were available for each testing subject.

E. RESULTS AND FINDINGS

The results of the usability tests focused primarily on the user's guides' usefulness, success, and shortcomings. The primary benefit of the tests were in the improvements to the document. Participants identified the following types of errors:

- typographical errors

- factual mistakes

- confusing layout and format

Improvements to the manual were indeed beneficial. The tests helped to refine the design of poorly composed sections of the user's guides. After observing the subjects and receiving their remarks upon completion of each of the tests, the guides were corrected and reformatted.

The information gathered as a result of the testing does not signify conclusive results due to "lack of rigor of the test methodology." [Zirinsky, 1986] However, the information was extremely useful in guiding the revision of the document design. The user's guides presented in Appendixes B and C are only prototypes and can be further refined, but can be used not to communicate with users at most levels of the audience continuum. Expert users of WordPerfect and dBase IV would probably find the user's guides least useful.

V. CONCLUSION

In concluding the results of this study, it appears that this researcher's intention to create user-friendly user's guides was achieved. The iterative methodology of the project development allowed the evaluation and refinement process to uncover errors and unclear sections. Changes were made that improved the manuals and helped make them viable tools for teaching and for reference.

The emphasis shifted during the study from creating guides that the writer thought was relevant, to creating a guide that revolved around user tasks. Successful documentation requires an ongoing dialogue between the documentation developer and the users.

Much of the difficulty people experience in learning a new computer system can be directly attributable to poor design. The egocentric style of designers must yield to humility, and designs adjusted to accommodate the users' skills, desires, and orientation. [Schneiderman, 1986] Designers need to understand that their design efforts may not always produce the desired effects for a particular audience, and be flexible enough to accept redesigning a system when necessary. The techniques for producing quality documentation will be of no use unless audience analysis is given high priority.

The ultimate test of the user's guides will be in the computer labs themselves and the classes in which they will be used. Refinements may still be required, corrections may have to be made, and more material may have to be added. Additionally, no computer program remains static for long, and trying to document the software has been likened to trying to change a tire on a speeding car. When new versions of the programs are released, new versions of the documentation will be required. To ensure consistency within the user's guides appended to this paper and among any future user's guides or lab manuals developed for the AS/IS Computer Labs, this researcher provided specifications, special graphics, and pertinent instructions with the AS/IS Computer Labs manager.

In conclusion, it can be said that the principles described and the findings noted in this study can be used by all documentation writers to improve their documentation. The benefits gained from the testing and reviewing of software documentation is a better understanding of user capabilities and improvements in design strategy. With the goals of the user clearly in focus, the production of higher quality, useful documentation can be achieved. The ultimate result is, however, in the acceptance and use of the user's guides by the people for whom it was designed.

APPENDIX A: AS/IS Computer Labs Software Directory

ROOM NUMBER:		I-224	I-224	I-250	I-158
NETWORK VENDOR & PROTOCOL: APPLICATION SOFTWARE	Version	IBM IBM TOKEN RING	3COM ETHERNET	ETHERNET BROADBND (PC NET)	IBM TOKEN RING
Hayes SMARTCOM II	2.1	X		X	X
SIMPC	2.1(c)	X		X	X
IBM PC 3270 Emulation	3.03	TN12,15 23,26-31		X	
WordPerfect	5.1	X		X	X
WordPerfect	5.0	X		X	X
WordPerfect	4.2				X
PeachText 5000	2.10		X		
LOTUS 1-2-3	2.01	X		X	X
STATGRAPHICS	2.6		X		
dBASE III+			X		
dBASE Administrator	1.1		X		
dBASE IV	1.0	X	X		
dBASE IV	1.1	X			
INGRES/DBMS	5.0/02a	X	X		
INGRES TUTOR			X		
EZ RATE Tariff 500-H				X	
Annualized Cost-of- Leaving Model (OSD)	1.0	X		X	
Universal Knowledge Management System	2.00.00		X		
IBM Storyboard Plus	1.01		X		
EtherMail	2.4		X		
EtherMenu	2.5(c)		X		
Microtek Int'l EyeStar	1.35	TN25			
IBM Virus Scanning Pgm	1.0	X	X		
Polaroid Palette PSAVER	2.0				X
Polaroid Palette for IBM PCs	3.1				TN 22M
1DIR	3.50	X	X	X	X
Force Analysis Simulation Model(FASM)	9.9			X	
Assembly	X				
BASIC	X				
C (Lattice)	X				
Turbo PASCAL	4.0	X			
Framework	X				
GRAMMATIK II	X				
Norton Utilities	X				
SYSTEM SOFTWARE					
IBM DOS	3.2		X		
IBM DOS	3.3	X		X	X
IBM PC LAN O/S	1.20	X		X	X
3COM Etherseries	2.4		X		

APPENDIX B: WordPerfect 5.1 Basic User's Guide

WordPerfect 5.1

A Basic User's Guide

For use on the

Administrative Sciences/Information Systems

(AS/IS) Computer Laboratories in

I-158, I-224, and I-250

September 1991

A C&P Product

Table of Contents

A local area network (LAN) is a group of microcomputers or other workstation devices located in the same general area and connected by a common cable. A LAN is designed to interconnect microcomputers, terminals, minicomputers, and other hardware, for the purpose of communicating among themselves and alternately with a host mainframe computer or public network.

The most common reason for developing a LAN is resource sharing. Networks allow the sharing of peripheral devices such as hard drives, printers, and scanners. Application programs such as spreadsheets, word processing, and communication packages can be shared so that multiple copies are not necessary. Databases can also be shared in such a way that multiple microcomputers can have access to a single database. This capacity for sharing hardware and software resources allows greater flexibility and cost savings in the use of expensive computer peripherals and software.

The basic components of a LAN are the server computer, the user computer(s), and the interconnecting cabling system. The server is usually a microcomputer that is specifically designated to act as the network server. The server performs only those functions required of a network server; it can only be accessed by users through their user computers. Server functions include repository of software programs, network management, printer and other peripheral device management, and database repository.

The user computer is normally a microcomputer or terminal, and is connected to the server by a cabling system. A simplified schematic of a typical connection is shown in Figure 1. One server can support more than one user computer, usually six to ten. The cabling system connecting the server and the users can be

Figure 1: LAN Schematic

present in a number of forms and configurations. Cabling can be twisted pair wire, coaxial cable, and fiber optic cable. Configurations include bus, ring, and star.

Logging on to a LAN as a user gives access to all the software on the server. When a software package is chosen, a copy of the software is downloaded to the user computer for execution. The user computer executes the software like a standalone computer, not accessing the server again unless a peripheral device, such as a printer, is needed. Further information on these and other local area network topics can be found in the references listed below.
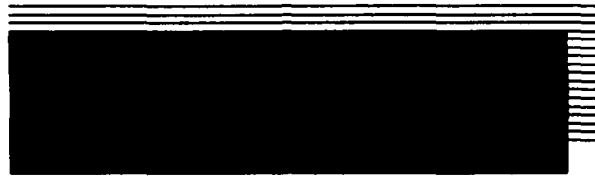
Suggested References

Berry, Paul, Operating the IBM PC Networks, Sybex, Inc., 1986.

Fitzgerald, J., Business Data Communications, John Wiley and Sons, Inc., 1990.

Madron, Thomas W., Local Area Networks: The Second Generation, John Wiley   and Sons, 1988.

Schatt, S., Understanding Local Area Networks, Howard W. Sams and Company, 1990.

This guide is primarily for those of you who are either new to WordPerfect and/or are new to working on a network and need to get WordPerfect running so you can create that first document. If you have used WordPerfect, or any other word processor, you are used to running the program from the subdirectory where the program files are stored. On a network the procedure is different. The program files are stored on the file server, where they are shared by other network users. If every network user stored document files in the WordPerfect subdirectory on the file server, it would be difficult to figure out which file belonged to which user and extremely difficult to locate files. Thus, users cannot edit, delete, or save files in the WordPerfect subdirectory; they need to save files to a floppy diskette.

WordPerfect is a text-oriented word processor, which means that as you create and edit your document, it appears on your screen as ASCII-coded characters (to see your document as it will be printed, you can select the WordPerfect **View document** feature. As soon as you boot up the system and select the WP51 start-up command from the menu, an initial WordPerfect start-up screen appears briefly. After that initial start-up screen, you will be presented with the document screen. On it, a default *status line* indicates the document window you are in (in WordPerfect, you can work on two documents at the same time) as well as the current page, line, and horizontal cursor position. The cursor always starts out in the upper-left corner. WordPerfect always starts you off in document 1 on page 1, and with top, bottom, left, and right margins of 1 inch. You issue commands to WordPerfect by pressing the function keys at the top (or left) of your keyboard, either alone or in combination with the **<Ctrl>**, **<Shift>**, and **<Alt>** keys. To see the formatting codes that are being used in your document, you press **Reveal Codes** (**<Alt-F3>**). You close this **Reveal Codes** window by pressing **<Alt-F3>** again. (**Note: <Alt-F3>** is also an example of a combination keystroke. It is done by first pressing the **<Alt>** key and, while the **<Alt>** key is held down, pressing the **<F3>** key. All combination keystrokes are done in this manner.)

$\boxed{\text{S}}$ tarting WordPerfect

1. Turn on your computer and log onto the network (follow the instructions provided at your computer). You will see, for the various applications available on the network, the ¡DIR menu with the batch file listings (files with .BAT extensions, that execute application programs).

2. Using the **Arrow-Down** key, move the select bar to the WP51.BAT file and the **<ENTER>** (also called the **<Return>**) key. (**NOTE: In I-250, you must hi formatted disk to insert in Drive A: before you can access WordPerfect. In you may use WordPerfect without your own disk; your work will be store C:\DATA (Drive C: in the DATA subdirectory). However, you will want to copy work onto your floppy disk to keep with you since the files on the C: drive m deleted at any time.)**

3. The WordPerfect document screen then appears. It is a blank screen exce the status information noted earlier); it is as if you were looking at a blank sheet of in a typewriter.

H *elp*

WordPerfect's online help is available any time you are working with the prc

1. To get help about the use of a particular function key or keystroke combi press **<F3> (Help)** and then press the key or keystroke combination to begin view information.

2. To get help about a particular feature/command by name, press **<F3>** follo\ the initial letter of the feature/command name (such as **<S>** to get help on S When a letter has more entries than will fit on one screen, type the letter again to another screen of entries. After locating the name of the feature on the Help s press the function keys indicated under the keystrokes column to obtain informatioi the feature's use.

3. WordPerfect's Help system is context sensitive, so when you are using a pa function, you can get more information about it by pressing **<F3>**.
4. To display a diagram of the function key assignments for WordPerfect, pres twice. A template of WordPerfect's function keys (Enhanced Layout) will appear screen. Press <1> and the IBM PC/XT (IBM Layout) will be displayed.

5. Press the **<ENTER>** key or **<Space Bar>** to exit the Help system.

R *etrieving a Document*

1. To start a new document, just begin typing in the document screen presented with at start–up.

2. To retrieve a document from a floppy disk:

a. Press **Retrieve <Shift-F10>**,

b. Type in the drive, path, and document name (e.g., A:\work\document.doc), and

c. Press **<ENTER>**.

3. If you do not remember the name of your document:

a. Press **<F5> (List Files)**,

b. Type the appropriate drive and path (e.g., A:\work*.*) to see the files in your work subdirectory on floppy Drive A:,

c. Highlight your document by using the arrow keys to move the cursor bar, and

d. Select the **Retrieve** option (type <1> or <R>).

4. **List Files <F5>** gives you an alphabetical listing of all files in the current or specified directory and allows you to perform common maintenance tasks: copy, delete, move, rename, print, and find.

5. If you retrieve a document while you are working on another document, you will see the prompt *Retrieve into current document?No(Yes)*. If you type <Y>, the document will be retrieved into the current cursor position in the document that is already on the screen.

C reating a WordPerfect Document

You can begin work immediately when you have the WordPerfect screen displayed on your monitor. You do not have to press **<ENTER>** at the end of each line since WordPerfect automatically word wraps for you. You do need to press **<ENTER>** at the end of each paragraph.

Moving the Cursor

The cursor is normally a small, flashing underscore that indicates the position of each character you type to the screen. If you want to edit some text, you have to move the cursor to the desired location in the document.

1. The four arrow keys on the right of the keyboard are the *cursor movement keys*. These arrow keys are collocated with the numbered keys on the numeric keypad. Press

the **<Arrow-Up>** key to move one line at a time up the page; the **<Arrow-Down>** key to move down one line, the **<Arrow-Left>** key to move left one character at a time, and the **<Arrow-Right>** key to move right one character. If you press a key down and hold it, the cursor will move continuously in the direction of the arrow.

2. To move the cursor one WORD at a time, press the **<Ctrl>** key and hold it down while you press the **<Arrow-Left>** key or the **<Arrow-Right>** key. Holding the **<Ctrl>** key down while pressing the **<Arrow-Up>** or **<Arrow-Down>** key will move the cursor to the beginning of the previous or next PARAGRAPH, respectively.

3. The **<Page-Up>** and **<Page-Down>** keys will move you to the beginning of the previous or the next PAGE, respectively.

4. The **Minus** key **<->** on the right side of your keyboard adjacent to the numeric keypad moves the cursor to the top of the SCREEN and the **Plus** key **<+>** next to the numeric keypad moves the cursor to the bottom of the SCREEN.

5. Other cursor movement techniques are as follows:

To Move	**<Key Sequence>**
To the beginning of a line	**Home, Home, Arrow-Left**
To the left edge of screen	**Home, Arrow-Left**
To the end of a line	**End**; or **Home, Arrow-Right**
To cursor's prior position	**Ctrl-Home, Ctrl-Home**
To the top of the page	**Ctrl-Home, Arrow-Up**
To the bottom of the page	**Ctrl-Home, Arrow-Down**
To a specified page (Go To)	**Ctrl-Home, {*page number*}, ENTER**
To the top of the document	**Home, Home, Arrow-Up**
To the end of the document	**Home, Home, Arrow-Down**

Inserting Text Into Your Document

Editing a document often requires adding new text. WordPerfect starts off in the default *Insert* mode. (If you see the word *Typeover* in the lower-left corner of your

screen, press the **Insert <Ins>** key located on the right side of your keyboard to return to the Insert mode.)

1. Position the cursor at the spot where you want to insert text.

2. Press the **<Arrow-Down>** key and WordPerfect will adjust the text to the correct margins.

Deleting Text

WordPerfect has many ways of deleting text, many of which are defined here:

To Delete	<Key Sequence>
Character by character	**Backspace** (deletes to left of cursor); **Delete <Del>** (deletes character or space the cursor is on).
Word by word	**Ctrl-Backspace**
Several words	**Escape <Esc>** *n* (*n* = number of words to the left of the cursor) **Ctrl-Backspace**
From the left of the cursor to the beginning of a word	**Home, Backspace**
From the cursor right to the end of a word (including the ending space)	**Home, Del**
To the end of a line	**Ctrl-End**
To the end of a page	**Ctrl-PgDn**
A sentence	**Ctrl-F4, S, D**
A paragraph	**Ctrl-F4, P, D**
A page	**Ctrl-F4, A, D**

Undeleting Text

WordPerfect 5.1 can restore any of the last three deletions at the cursor's position. If WordPerfect is not carrying out a command, the **Cancel** key **<F1>** functions as the **Undelete** key. The following prompt appears along with the most recently deleted text:
Undelete: 1 Restore; 2 Previous Deletion: 0
Choosing *Restore <1> or <R>* restores the displayed text to your document; choosing *Previous Deletion <2> or <P>* displays the text that was deleted prior to that deletion. The last three deletions can be displayed and restored. After the third most recently deleted text is displayed, selecting *Previous Deletion* displays the first deletion again. Selecting *Restore* restores the displayed deletion to your document.

Using the Typeover Feature

With the *Typeover* feature, you can enter replace existing text without pushing the rest of the sentence to the right. Press the **Insert <Ins>** key until the *Typeover* prompt appears in the lower–left corner of your screen. The *Insert* mode is now off.

F unction Keys
_____

The function keys are listed here alphabetically and sequentially by F–key.

Alphabetical Listing

Function	<Key Sequence>	Brief Description
Block	**Alt-F4**	Defines a block of text on which you can then perform any number of operations.
Bold	**F6**	Prints selected text in boldface or doublestrike.
Cancel/ Undelete	**F1**	Terminates almost any command being carried out/ Restores up to three previous deletions.
Center	**Shift-F6**	Centers text on a line between left & right margins.
Columns/ Tables	**Alt-F7**	Format your text using columns/create tables.
Date/ Outline	**Shift-F5**	Inserts the current date as text or code/create an outline of your document.
End Field	**F9**	End of field code in a record (used in merging)

Exit	**F7**	Quits WordPerfect or current screen.
Flush Right	**Alt-F6**	Aligns your text flush with the right margin setting.
Font	**Ctrl-F8**	Allows you to change the size or appearance of the current fonts used in your document.
Footnote	**Ctrl-F7**	Allows you to add footnotes that appear at the bottom of the page or endnotes that appear at a place of your choice in the document.
Format	**Shift-F8**	Controls most aspects of the document format using 4 submenus: Line, Page, Document, and Other. Used to set margins.
GoTo	**Ctrl-Home**	Moves the cursor to a specific character, page, or text column, or to the previous cursor position.
Graphics	**Alt-F9**	Allows you to combine graphics created by other programs with the text of your document or to draw rules (lines) in the document.
Hard Page	**Ctrl-ENTER**	Ends a page at the discretion of the user.
Help	**F3**	Gives you on–line help about a function key, function, or a WordPerfect command.
Indent	**F4**	Sets a temporary left margin and aligns all text to this indent until you press **<ENTER>**.
Indent	**Shift-F4**	Sets temporary left and right margins and aligns all text to these indents until you press **<ENTER>**.
List	**F5**	Displays an alphabetical listing of all files in the current directory; allows common maintenance tasks: retrieve, delete, move/rename, print, copy, text in, and look
Macro	**Alt-F10**	Executes a defined macro.
Macro Define	**Ctrl-F10**	Defines a macro (begins recording keystrokes which can be replayed any time).

Margin Release	**Shift-Tab**	Moves the cursor one tab stop to the left.
Mark Text	**Alt-F5**	Compares documents, removes redline markings and strikeout text, and creates automatic references, master documents, indexes, lists, tables of authorities, and tables of contents.
Merge Codes	**Shift-F9**	Designates a field from the secondary file to be merged in the primary file.
Merge/Sort	**Ctrl-F9**	Performs a merge of data stored in lists in a secondary document into the appropriate places in a primary document.
Move	**Ctrl-F4**	Allows you to move, copy, or delete a sentence, paragraph, or page. You can then move to another place in the document and retrieve the text.
Print	**Shift-F7**	Allows you to print a document or page. Also allows other functions, such as view document.
Replace	**Alt-F2**	Allows you to select any sequence of characters or codes and globally change it to something else.
Retrieve	**Shift-F10**	Retrieves a document on disk or the last text that was cut or copied.
Reveal Codes	**Alt-F3**	Splits the screen and allows you to see the hidden codes, which instruct the printer on how to format text and graphics in the document.
Save	**F10**	Saves a document on disk under the name you assign.
Screen	**Ctrl-F3**	Allows you to draw straight lines and boxes in the document, turn off/on automatic screen writing, and split the document screen into two windows.
Search	**F2**	Locates the next occurrence in the document of specified text or formatting codes.
Search	**Shift-F2**	Performs a backward (reverse) search.

Setup	**Shift-F1**	Allows you to change many of WordPerfect's default settings.
Shell	**Ctrl-F1**	Allows you to exit temporarily to DOS.
Spell	**Ctrl-F2**	Allows you to check the spelling of a word, a block of text, or an entire document.
Style	**Alt-F8**	Allows you to store sets of formatting commands that can be applied to various parts of your document.
Switch	**Shift-F3**	Converts defined block to all UPPERCASE or all lowercase letters. Switches between the Doc 1 and Doc 2 editing screens.
Tab Align	**Ctrl-F6**	Aligns text on or around the next tab stop using the decimal/align character.
Text In/Out	**Ctrl-F5**	Allows you to retrieve a DOS text (ASCII) file into WordPerfect; to save a document as a DOS text or other formats (such as previous versions of WordPerfect); to create document comments; and to assign passwords to documents.
Thesaurus	**Alt-F1**	Allows you to look for synonyms for any word in the text of your document.
Underline	**F8**	Begins underlined text or underscores selected portions of text.

Sequential F-Key Listing

	<Key>	**<Alt>**	**<Shift>**	**<Ctrl>**
F1	Cancel	Thesaurus	Setup	GoTo DOS
F2	>Search	Replace	<Search	Spell

F3	Help	Reveal Codes	Switch	Screen
F4	>Indent	Block	>Indent<	Move
F5	List	Mark Text	Date/Outline	Text In/Out
F6	Bold	Flush Right	Center	Tab Align
F7	Exit	Columns/Table	Print	Footnote
F8	Underline	Style	Format	Font
F9	End Field	Graphics	Merge Codes	Merge/Sort
F10	Save	Macro	Retrieve	Macro Define

B lock Operations

The **Block** command **<Alt-F4>** is used to highlight (mark) a section of text for use with other WordPerfect commands. To mark a block of text:

1. Position the cursor at the beginning of the block and press **Block (<Alt-F4>)**. The message *Block on* begins blinking at the bottom–left of your screen.

2. Position the cursor at the end of the block (as you move the cursor, the included text will be highlighted), then press **Block (<Alt-F4>)** again.

3. Select the operation you want applied to the block. You can choose from the menu items at the bottom of the screen or use a function key or combination of function keys, as summarized below:

<u><Key Sequence></u>	<u>Action with Block On</u>
F1 (Cancel)	Cancel block
Alt-F2 (Replace)	Replaces in block
Ctrl-F2 (Spell)	Checks block
Shift-F3 (Switch)	Changes block to all uppercase or lowercase
Ctrl-F4 (Move)	Cuts, copies, or moves block; Cuts/copies column or rectangle
Alt-F5(Mark Text)	Marks for ToC, list, index, paragraph numbering
F6 (Bold)	Bolds block
Alt-F6 (Flush Right)	Moves block flush with right margin
Shift-F6 (Center)	Centers block
Shift-F7 (Print)	Prints block
F8 (Underline)	Underlines block
Shift-F8 (Format)	Protects block
Ctrl-F9 (Merge/ Sort)	Sorts block
F10 (Save)	Saves block in a new file

P *rinting and Viewing a Document*

From the screen, you can print the entire document, a single page, or a marked block of text. It's a good idea to view your document before you print, though, to make sure it looks like you want it to look; you will save costly printer paper and time by first previewing your document.

Printing

The system boot disk provided automatically captures a printer port for your workstation when you log in, thereby ensuring that print jobs go to the file server queue and subsequently print on network printers. You do not have to manually select a port.

1. Press the **Print** key (**<Shift-F7>**). WordPerfect presents you with a Print menu.

2. Press the **<1>** or **<F>** to print the entire document (the cursor may be placed anywhere in the document to print the whole document), or the **<2>** or **<P>** to print one page (it will print the page the cursor is on at the time). You are returned to your document screen, and the print job goes to the file server print queue where it waits its turn to be printed. A separator page with the network operating system's name, your login name, and the print job number will be printed first so you can identify your print job.

3. To print a block of text from the screen, move the cursor to the first character of the block you want to print; press **<Alt-F4> (Block)**. The message *Block* flashes in the lower-left corner of the screen.

4. Move the cursor to the character space immediately after the last character of the block of text you want to print. Press **<Shift-F7> (Print)**.

5. Press **<Y>** for yes to the prompt *Print block? No (Yes)*.

Viewing

You can view the document pages as they will appear when printed on paper, including graphics, footnotes, page numbers, line numbers, headers, footers, and justification. However, on the Token Ring network in I-224, three computers (TN12, TN15, and TN23) will not allow you view without a few extra steps; that's because WordPerfect is preset for EGA monitors whereas these three stations have CGA monitors. For instructions, see the footnote[1].

---

[1] To be able to view your document from TN12, TN15, or TN23 on the Token Ring network in I-224, you need to reset the default monitor selection by following these steps:
1) Press **<Shift-F1> (Setup)**;
2) Select **<2> (Display)**; (continued next page)

1. Put the cursor on the page you want to view.

2. Press **<Shift-F7> (Print)** and select **<6>** or **<V>** (for View document).

3. Press **<1> (100%)** to view the document at its actual size;
   press **<2> (200%)** to view at twice its actual size;
   press **<3> (Full Page)** to view the whole page at once;
   press **<4> (Facing Pages)** to view the current page and its facing page
(even—numbered pages are shown on the left, odd—numbered pages on the right).

4. Press **<PgUp>** or **<PgDn>** or **<Ctrl-Home> (GoTo)** to view other pages in the document.

5. Press **<F7> (Exit)** to return to the document screen.

S aving a Document

Once you create or modify your document, you will want to save it on your floppy diskette. You can save the document either while you are still working on it or when you are finished with and are ready to exit WordPerfect. It is, however, a very good idea to SAVE often!! It only takes a few seconds and can save HOURS of retyping.

Saving without Exiting

You should save your document every 10–15 minutes in case there's a power surge or failure affecting your workstation or the file server. If you leave your workstation for any reason while working on a document, make it a habit to save your work first.

1. Press **Save (<F10>)**. You will see this prompt in the lower–left corner of your screen: *Document to be Saved:*.

   a. If you retrieved the document from your floppy diskette or saved
   the file previously, WordPerfect will provide the drive, path, and
   document name (e.g., A:\WORK\DOCUMENT.DOC) as a
   default.

---

3) Select **<2>** (Graphics Screen Type);
4) Select **<2>** (Auto-select);
5) Press **<ENTER>**;
6) Press **<ENTER>**, then follow the instructions for viewing in
the body of the guide.

(1) To save the file under the same name, press **<ENTER>** and respond **<Y>** to the prompt *Replace A:\WORK\DOCUMENT.DOC? No(Yes)* to indicate that you <u>do</u> want to replace the original version of the file with the edited version. The default setting is No, which allows you to leave the existing file intact and save the new version under another name.

(2) If you want to rename the file in order to keep two versions of a document, press **<ENTER>** to accept the No setting. WordPerfect will allow you to enter a new file name (including drive and path designations). Type the new name or edit the existing name, then press **<ENTER>**.

b. If you newly created the document and have not saved the file previously, WordPerfect will prompt you for a file name. Type in a drive and path designation, and a file name of up to eight characters with an optional three–character extension (characters after the dot); for example, A:\WORK\DOCUMENT.DOC. Then press **<ENTER>**. The file will be saved on your floppy disk.

2. Once the document has been saved, you will be returned to the document screen.

Saving and Exiting WordPerfect / Clearing the Screen

When you are finished with your word processing session, you will want to save your latest document and exit WordPerfect.

1. Press **<F7>**, the **Exit** key.

2. You will see the prompt, *Save document? Yes(No)*. Press **<ENTER>** or **<Y>** to accept the default setting of Yes. If you want to abandon the document and any changes you have made to it, press **<N>**. You can then exit without saving the document.

3. After you have pressed **<ENTER>**, WordPerfect prompts you for the name of the document.

a. If you have not saved the document before, type in the directory path and document name and press **<ENTER>**. (Remember to save your work to your formatted floppy disk before leaving.)

b. If you have already saved the document at least once, the prompt will contain the directory path and document name.
(1) If you want to save it under the same name, simply press

<ENTER>. WordPerfect will respond with the prompt, *Replace A:\WORK\DOCUMENT.DOC?No(Yes)*. If you change your mind and decide to rename the document, press **<ENTER>**. Otherwise, type **<Y>** for Yes. (2) If you want to save the document under a new name, type it in or edit the existing name and press **<ENTER>**.

4. After the document is saved, you will see this prompt, *Exit WP?No(Yes)*.

   a. The default response is No, so you can press **<ENTER>** or **<N>** to **CLEAR THE SCREEN** and begin creating a new document or retrieve another document for editing.

   b. If you decide you do not want to have your document cleared from the screen, press Cancel (<F1>) to retain it and return to editing its text.

   c. If you answer **Yes** to exiting WordPerfect, you will be returned to the 1DIR menu to select another application or log off the system.

Bibliography

Harvey, Greg and Nelson, Kay Yarborough, <u>WordPerfect 5.1 Instant Reference</u>, Sybex, Inc., 1990.

Simpson, Alan, <u>Mastering WordPerfect 5.1</u>, Sybex, Inc., 1990.

Index

APPENDIX C: dBASE IV 1.1 Basic User's Guide

dBase IV 1.1

A Basic User's Guide

For use on the

Administrative Sciences/Information Systems

(AS/IS) Computer Laboratories in

I-158, I-224, and I-250

September 1991

A C&P Product

Table of Contents

A local area network (LAN) is a group of microcomputers or other workstation devices located in the same general area and connected by a common cable. A LAN is designed to interconnect microcomputers, terminals, minicomputers, and other hardware, for the purpose of communicating among themselves and alternately with a host mainframe computer or public network.

The most common reason for developing a LAN is resource sharing. Networks allow the sharing of peripheral devices such as hard drives, printers, and scanners. Application programs such as spreadsheets, word processing, and communication packages can be shared so that multiple copies are not necessary. Databases can also be shared in such a way that multiple microcomputers can have access to a single database. This capacity for sharing hardware and software resources allows greater flexibility and cost savings in the use of expensive computer peripherals and software.

The basic components of a LAN are the server computer, the user computer(s), and the interconnecting cabling system. The server is usually a microcomputer that is specifically designated to act as the network server. The server performs only those functions required of a network server; it can only be accessed by users through their user computers. Server functions include repository of software programs, network management, printer and other peripheral device management, and database repository.

The user computer is normally a microcomputer or terminal, and is connected to the server by a cabling system. A simplified schematic of a typical connection is shown in Figure 1. One server can support more than one user computer, usually six to ten. The cabling system connecting the server and the users can be present in a number of forms
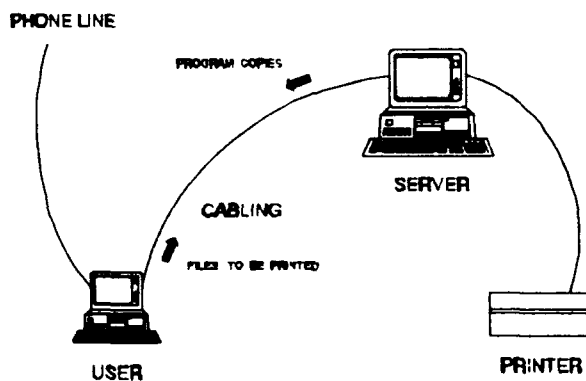

Figure 1: LAN Schematic

and configurations. Cabling can be twisted pair wire, coaxial cable, and fiber optic cable. Configurations include bus, ring, and star.

Logging on to a LAN as a user gives access to all the software on the server. When a software package is chosen, a copy of the software is downloaded to the user computer for execution. The user computer executes the software like a standalone computer, not accessing the server again unless a peripheral device, such as a printer, is needed. Further information on these and other local area network topics can be found in the references listed below.

Suggested References

Berry, Paul, Operating the IBM PC Networks, Sybex, Inc., 1986.

Fitzgerald, J., Business Data Communications, John Wiley and Sons, Inc., 1990.

Madron, Thomas W., Local Area Networks: The Second Generation, John Wiley and Sons, 1988.

Schatt, S., Understanding Local Area Networks, Howard W. Sams and Company, 1990.

dBase IV is a database management system which lets you create many different types of databases, make data additions and deletions, search for records with a variety of search criteria, sort data into multiple formats, print mailing labels, generate reports, and, if you work with numerical data, perform various mathematical computations. This guide is designed to familiarize the beginning user of dBase IV with a few of the program's basic operations. More comprehensive treatments are available which cover more of the many features dBase IV offers.

Like other applications, the dBase IV program is stored on the file server hard disk in its own subdirectory. Database files which you create and work with should be stored on your floppy disk or a class subdirectory, which your instructor will explain.

dBase IV stores information in *database files* which consist of *records* (or *rows*). Each record is divided into separate *fields* (or *columns*). Figure 2 depicts a dBase IV database file containing names and addresses. *Forms*, such as the most basic Edit screen, are used to gather information on the computer screen and that information is stored in the database file. *Reports* are derived from the information in the database and can be printed, in the format you specify, on your computer screen or printer. dBase IV can also pull together information from separate database files into a single printed report. In managing a database, you execute a few basic tasks:

- ADD new information
- CHANGE information
- DELETE old information
- SORT information into some useful order (like alphabetical order)

Figure 2. A dBase IV Database File

- SEARCH for particular types of information (like a specific address

90

or names of all people with San Diego addresses)
– CALCULATE totals, subtotals. counts, and averages

dBase IV offers an optional *command-driven* technique for managing databases, where you type commands at a dot prompt screen rather than select options from pull-down menus. This guide does not deal with the command–driven technique. However, if you inadvertently get to the dot prompt, you can get back to the Control Center by typing ASSIST and pressing **Enter**, or just pressing the **F2** key.

S | tarting dBase IV

1. Turn on your computer and log onto the network (follow the instructions provided at your computer). You will see the IDIR menu with the batch file (files with .BAT extensions, that execute application programs) listings for the various applications available on the network.

2. Using the **Arrow-Down** key, move the select bar to the DBASE4.BAT file and press the **Enter** (also called the **Return**) key.

3. The dBase IV logo and copyright notice appear on the screen, along with the message "*Press Enter to assent to the License Agreement and begin dBase IV.*"

4. Press **Enter** to start the program or wait a few moments and dBase IV will display the Control Center.

T | he Control Center

After the dBase IV logo and copyright notice disappear, the Control Center appears on your screen. The operations performed by dBase IV are initiated from the Control Center. The Control Center provides a user interface based on pull–down menus, rather than commands, for interacting with dBase IV. It is the primary user–assistance feature for non–programmers in dBase IV. The five major parts of the Control Center are: the *menu bar*, the *catalog line*, the *file panels*, the *current file description line*, and the *navigation line*. The *pointer* is a highlighted bar that is moved by the **Arrow** keys to the menu options and files; to select an option from the Control Center, position the pointer and press **Enter**.

The Control Center displays the names of files in the current catalog in *the panels*, which are labeled *Data, Queries, Forms, Reports, Labels, and Applications.* The names of various files from other users may appear in these panels, but they will not affect your work.

The Menu Bar

At the upper-left corner of the Control Center is the horizontal *menu bar*, which offers the options Catalog, Tools, and Exit. Each option has a pull-down menu associated with it.

1. Pull-down menus can be accessed by pressing the **Menu** key (**F10**) or by holding down the **Alt** key and typing the first letter of the option you want.
   a. Shaded options here are not available at the moment, usually because they make no sense in the current situation.

   b. Bulleted options (those with arrows/right-pointing triangles) have submenus associated with them.

2. To leave the pull-down menus at any time, press the **Esc** key until the menus disappear.

The following keys are used to navigate the pull-down menus. You may want to try some of these keys to see how they work.

KEY	EFFECT
Enter	Selects the currently highlighted option
Arrow-Left, -Right	Moves to menu-bar option on left or right
Arrow-Up, -Down	Moves up or down to next available pull-down menu option
PgUp, Home	Moves to first available pull-down menu option
PgDn, End	Moves to last available pull-down menu option
First letter of any option	Selects that option
Esc	Backs up to prior menu or Control Center

The Catalog Line

The catalog line is centered beneath the title of the Control Center screen, and displays the name of the current catalog. If no catalog has been created, the catalog line

will read *UNTITLED.CAT.* A catalog is a grouping of related files in a database, such as storing accounts receivable information in one catalog and inventory information in another.

The File Panels

The file panels of the Control Center are the six vertical rectangles in the center of the screen labelled Data, Queries, Forms, Reports, Labels, and Applications. They contain the names of the various types of database files in the current catalog. For instance, the Data panel holds the names of database files that contain data; the Forms, Reports, and Labels panels hold the names of formats used to display forms and to print reports and mailing labels; and the Queries and Applications panels hold the names of query files and dBase IV application programs (dBase IV applications will not be covered by this manual). Each panel also includes the option <create>, which allows you to create a new database file, form, report, or query; it is placed into the current catalog.

The Current File Description Line

The *File:* and *Description:* section, just beneath the file panels, provide the name of the currently selected file and a brief description of the file's contents. If <create> is highlighted instead of a file name, this area displays "New file" and "Press ENTER on <create> to create a new file."

The Navigation Line

At the bottom of the screen is the navigation line. It displays the commands that are available for the option currently selected. As an example the navigation line displayed at the Control Center shows:

Help: F1	Pressing **F1** always displays a Help screen
Use: ⬅ **Enter (Return)**	Use the highlighted item
Data: F2	Display data in Browse mode (Pressing **F2** a second time displays data in Edit mode)
Design: Shift-F2	Displays the Database Design screen
Quick Report: Shift-F9	Displays the Print menu.
Menus: F10	Activates the menu bar

$\boxed{G}$ etting Help

You can use the **Help** key (**F1**) at any time to get help with dBase IV. The help screens are *context sensitive*, which means they provide help that is relevant to the operation you are currently performing (whichever operation is currently highlighted). After pressing **F1**, a help screen (or window) will appear on the screen. The navigation line at the bottom of the screen will tell you the keys to use to navigate the help screens, usually **F4** for *next screen* and **F3** for *previous screen*. The **Arrow-Left** and **Arrow-Right** keys will move you among any help options offered in the help window.

The help system's Table of Contents can be accessed by highlighting the *CONTENTS* option and pressing **Enter**. You'll see a list of topics concerning database files.

<u>HELP OPTION</u>	<u>BRIEF DESCRIPTION</u>
CONTENTS	Displays a table of contents for the current topic. When a table of contents is displayed, you can use the **F3** (*More General*) or **F4** (*More Specific*) key to change to more general or more specific tables of contents.
RELATED TOPICS	Displays a list of topics related to the current topic.
BACKUP	Scrolls back to the previous screen. This option only appears if there is a previous screen.
PRINT	Prints a copy of the current help window.

When you are finished with the help system, return to the Control Center by pressing the **Esc** key.

$\boxed{C}$ reating and Saving a Database Structure

Each piece of raw data—a phone number, name, street address, inventory code, and so on—is placed into its own field. Several fields of related data are grouped together to create a record. Records are compiled into a database file, and related database files are organized into a catalog.

Creating a Catalog

A catalog name is a DOS filename of up to eight letters or numbers. An extension is not included because dBase IV assigns the extension .CAT to all catalogs.

1. To open the Catalog menu, press **F10**, the **Menu** key. The pointer moves to "Catalog" on the menu line, and the Catalog menu drops down.

2. The pointer is on the "*Use a different catalog*" option. Press the **Enter** key. The Catalog box appears in the upper-right corner of the screen. You may see a list of previously created catalogs. You may select one of these and press **Enter** to display a list of that catalog's files, or

3. Place the pointer on <create> and press the **Enter** key to create a new catalog.

4. Type the name of your new catalog and press **Enter**. You are returned to the Control Center, where the new catalog name will appear in the catalog line.

Defining Data Fields

A data field contains one specific item of data. The Database Design screen is where you define each field in your database structure.

1. The pointer should be positioned on <create> in the Data panel. Press the **Enter** key. The Database Design screen appears. (Notice the message and navigation lines at the bottom of the screen. dBase is prompting you to enter the first field name. Also, a new line of highlighted information, called a **status** line, appears at the bottom of the screen.)

2. To define the fields you want to use in your records, you must fill in the options indicated by the six field panel headings: *Num, Field Name, Field Type, Width, Dec,* and *Index.*

    a. Num: The Num column tells you the field number you are working with. It is defined by dBase and cannot be specified.

    b. Field Name: You can enter any name that will identify the type of information to be entered in that field, up to ten characters (either upper- or lower-case). It can contain only letters, numbers, and underscores and the first character must be a letter. Each field name in a database file must be unique.

    c. Field Type: Here, you must specify what kind of information the field is going to hold. There are six options, which you can select by pressing the space bar until your choice appears in the place where "Character" originally appears. "Character" is the default setting since most database fields are of character type. You can also type the first letter of a choice to select it. The six types are:

1) Character type: all letters, punctuation marks, special symbols, and any numerals, like zip codes and phone numbers. (Mathematical calculations cannot be performed on numbers in this field type.)

2) Numeric type: numbers that will later require mathematical calculation or that have a fixed decimal point, such as dollar amounts.

3) Float type: numbers that have a floating decimal point, values that sometimes require a decimal point and sometimes do not, and negative numbers.

4) Date type: dates such as 9/26/91.

5) Logical type: true/false or yes/no entries.

6) Memo type: This field type can include large amounts of texts, such as abstracts, comments, or lengthy descriptions.

Select the field type and press **Enter**.

d. Width: Here you must specify the maximum number of letters, numbers, or other characters that you plan to put into the field. The Logical, Memo, and Date fields are automatically assigned a width by dBase IV, so you have to specify a width only for Numeric, Float, and Character fields.

Specify the number of characters, then press **Enter**.

e. Dec: If you entered N (for Numeric) as the field type, the cursor will stop in this column. You must enter here the number of decimal places you want in the field. For example, for monetary figures you would type **2** to indicate that you want two decimal places.

Type in the number of decimal places and press **Enter**.

f. Index: This column is used to control how dBase orders your records. A Y(es) or N(o) indicates whether or not the field should be indexed. If this option is changed to **Y**, dBase will assign the name of the field as the name of the index and actually create the index when you save the database structure. For more on indexing, see Indexing, page 13.

Type **Y** or **N** and press **Enter**.

3. The cursor is now positioned to define the second field. Following the procedure described above, enter the remaining field information for your record and save this database file structure as described in the "Saving the Database Structure" section below.

Making Changes and Corrections

At this point, if you notice an error in your file structure or wish to make a change, you can use the keys described here, to change your database file structure. The keys won't always work if you attempt to leave a field that has incomplete or invalid data. You may

have to experiment with several keys or type **Ctrl-U** to completely delete an incomplete or invalid field to get the cursor moving in the direction you want.

KEY	EFFECT
Enter	Completes an entry and moves to the next column or row
Arrow-Left, -Right	Moves cursor one character to the left or right
Arrow-Up, -Down	Moves highlight up or down one row
Home	Moves cursor to the first column in the row
End	Moves cursor to the last column in the row
Ctrl-End	Saves changes and exits the Database Design screen
Tab	Moves cursor one column to the right (only if valid information is already in the present column)
Shift-Tab	Moves cursor one column to the left (only if valid information is already in the present column)
Backspace	Moves cursor one space back, erasing along the way
F1 (Help)	Displays Help screen
F2 (Data)	Switch to browse or edit screen
Ctrl-N	Inserts a blank field between two existing fields
Ctrl-U	Deletes the current field
Ctrl-W	Saves changes and exits the Database Design screen
Shift-F2 (Design)	Transfer to Query Design screen
Esc	Exits without saving changes and returns to previous screen

Saving the Database Structure

1. From the Database Design screen, pull down the Layout menu by pressing the **F10** key.

2. Move the pointer to _Save this database file structure_. Press the **Enter** key. The _Save as:_ box appears.

3. Type in the name of your database file (dBase IV will add the file extension .DBF to each database file) and press **Enter**. The database file structure is now saved, ready to be filled with the actual data for each record.

4. To exit the Database Design screen and return to the Control Center, press the **F10** key. The Layout menu pops down again but displays different options this time.

5. Press the **Arrow-Right** key to move the pointer to Exit. Notice how each menu box pops down as the pointer moves from one menu item to another.

6. At the Exit menu, the pointer should be on the _Save changes and exit_ option. Press the **Enter** key. You are returned to the Control Center.

W orking with the Database

Placing information in your database files begins with creating records and entering data into the fields from the Edit screen. The records can also be edited on the Edit screen. The Edit screen looks like a blank form for a single record and is useful when you want to concentrate on one record at a time (custom forms can be created for entering and editing data, but are not covered in this guide). You can move from field to field using the following keys, which are useful in the Browse screen also:

KEY	EFFECT
Enter	Completes entry and moves to next field
Arrow-Left, -Right	Moves the cursor left or right one character
Arrow-Up, -Down	Moves the cursor up or down
PgUp, PgDn	Moves up or down one record on Edit screen or one screenful on Browse screen

Home	Moves to first filed on Browse screen or first character in current field on Edit screen
Ctrl-Home	Moves from a memo–field marker into an editing window to add or change a memo
End	Moves to last field on Browse screen or end of current field on Edit screen
Ctrl-End	Saves changes and exits to previously used screen
Tab	Moves to next field, or indents paragraph in word–wrap editor
Shift-Tab	Moves to previous field, or outdents paragraph in word–wrap editor
Ins	Switches between Insert and Overwrite modes
Backspace	Moves left one character, erasing along the way
Del	Deletes one character over cursor
F1 (Help)	Displays help
F2 (Data)	Toggles between Browse and Edit screens
F3 (Prev)	Scrolls back to previous field
F4 (Next)	Scrolls to next field
F10 (Menu)	Accesses pull–down menus
Ctrl-Y	Deletes all characters to right of cursor
Shift-F8 (Ditto)	Carries data from same field in previous record to current record
Esc	Leaves current screen without saving changes to last–edited record

Entering Data Into Records

1. From the Control Center, place the pointer on the database file in the Data panel in which you want to place information. Press the **Enter** key.

2. Move the pointer to *Display data* and press **Enter**. The fields you created for your database structure will appear on the screen. This is the Edit screen as indicated by the word **Edit** at the left end of the **status** line.

3. Fill in each field with the appropriate data, pressing the **Enter** key after each entry. When you have filled in all the fields, you will have completed a record and dBase moves to the next blank record.

Editing a Record In the Edit Screen

From the blank record you are in, press the **PgUp** key to move to a previous record. To move to the field you want to edit, use the **Arrow-Up** and **Arrow-Down** keys. You can type over existing text (to insert, press the **Insert (Ins)** key).

Exiting the Edit Screen

1. Press the **PgDn** key until you reach the last record and see the following prompt at the bottom of the Edit screen:

=== > Add new records?(Y/N)

2. Press **Y** and a blank set of fields appears.

3. Press the **Enter** key and you are returned to the Control Center.

Editing and Adding Records In the Browse Screen

1. From the Control Center, press the **F2** key. This is the Browse screen (see left end of **status** line; if it says **Edit**, press **F2** once more). To edit an existing record, simply move the cursor to the field that needs changing and type in the correct information.

2. To add a record, press the **Arrow-Down** key until the following message appears at the bottom of the Browse screen:

=== > Add new records?(Y/N)

3. Press **Y**. The pointer moves to the first field of the next blank record.

4. Type in the data in the fields. pressing **Enter** after each entry.

The keys listed on pages 12 and 13 can be used to navigate the Browse screen.

Undoing an Edit

Occasionally, you may change or erase a field's contents accidentally. Before you move to a new record, you can undo an edit.

1. Highlight the *Records* option on the menu bar.

2. Select *Undo Change to Record* and the previous contents of the record will be restored.

Memo Fields

A memo field marker appears as a highlight with the word *memo* on the Edit and Browse screens. When the memo field contains information, the marker is shown in uppercase letters.

1. To open the memo marker, move the cursor to it and press **Ctrl-Home** or **F9 (Zoom)**.

2. After editing the memo field, save your work and return to the form or Edit screen by selecting *Save Changes and Exit* from the Exit pull-down menu or by pressing **Ctrl-End**.

Viewing a List of Records

On the Edit screen you are limited to viewing one record at a time. With the Browse mode, however, you can view a screen full of records.

1. From the Control Center, move the pointer to a database file name in the Data panel.

2. Press the **F2** key. The Browse screen should appear. If the Edit screen appears, press **F2** again. The records are listed down the screen and field columns are displayed across the screen. If you see only one record, press the **PgUp** key to pull other records into view. (**Note:** You may not see all the fields at once. Use the **Tab** key to view hidden columns.)

Marking Records for Deletion

Before you delete a record it must be *marked* for deletion. From the Browse screen, you can mark one or more records at a time using either of two different methods.

Method one:

101

1. Move the pointer to the first field of the record to be deleted and press **F10**. The Records menu appears.

2. Move the pointer to the *Mark record for deletion* option and press the **Enter** key.

3. The record is now marked. Notice the **Del** at the right end of the **status** line.

Method two:

1. From the Browse screen, move the pointer to the record to be deleted, then press **Ctrl-U**.

2. The record is now marked and **Del** is displayed on the **status** line.

Unmarking a Record for Deletion

There are two methods to unmark a record for deletion.

Method one:

1. Place the pointer on the marked record and press the **F10** key. The Records menu appears.

2. Move the pointer to the *Clear deletion mark* option and press the **Enter** key.

3. The **Del** on the **status** line disappears, indicating the record is no longer marked for deletion.

Method two:

1. Move the pointer to the record to be unmarked, then press **Ctrl-U**.

2. The record is now unmarked and **Del** disappears from the **status** line.

Deleting Marked Records

When you delete records from the database file, dBase IV shifts the remaining records to occupy the space on the disk made vacant by the deleted records. This process is referred to as *packing the database.*

1. From Control Center screen, press **Shift-F2**. The Database Design screen appears and the Organize menu pops down.

2. Move the pointer to *Erase marked records.* Press **Enter.**

3. A box pops out with the question, *Are you sure you want to erase all marked records?* Select **Yes** and the database is packed. You are returned to the Database Design screen. The records have been deleted from the database.

4. To exit, press the **F10** key, move to the Exit menu, and press the **Enter** key.

S | orting a Database

To sort the information in your database in a useful order, such as alphabetically or zip-code order, dBase IV offers two methods. The fastest and most efficient method is by *indexing.* The second way is to make a *sorted copy* of a database file. This guide will only cover the first method—indexing.

Indexing

A dBase IV index is a sorted list of items in a database file. When you activate the index, dBase automatically displays the records in the sort order specified by the index. The actual records in the database are still in their original order; the index just "tells" dBase the order in which to display information on the screen or printer. Also, dBase can use the index to quickly locate an item of information in the database.

You can create or modify an index from the Database Design screen at any time, whether or not the database contains records. You can create up to 47 indexes for any given database. dBase IV stores all the indexes for a given database in a file with the same name as the database, but with the extension .MDX. When you add, change, or delete records, dBase updates the indexes automatically.

1. Highlight the database name in the Data panel of the Control Center and then press **Shift-F2** to invoke the Database Design screen.

2. Select *Create New Index* from the Organize pull-down menu and press **Enter.** A submenu appears asking for information about the index.

3. Press **Enter** to select *Name of Index.* Type the index name (also called the *tag*) and press **Enter.** You can assign any name to the index, following the same basic guidelines as for creating field names (see Field Name, page 13).

4. Now select the *Index Expression* option and press **Enter.** Type the <index expression> and press **Enter.** The index expression is a single field name for a simple index. You cannot index on logical or memo fields.

5. Select the *Order of Index* option (skip the **FOR** clause; it will not be covered). You can choose **Ascending** (lowest to highest) or **Descending** (highest to lowest) by pressing the Space bar.

6. Press **Ctrl-End** to save your work (the last entry will not be covered either).

dBase IV will create the index and show its progress on the screen. When it is done, the Database Design screen reappears.

Sorts within Sorts

If sorting your database on a single field is not sufficient, such as when you want to sort records in last-name order and then in first-name order for those with identical last names, you need to perform a sort within a sort. To do this, follow steps 1–6 above, under indexing. However, use an index expression that lists the fields to sort on in priority order with a plus sign (for character types) between each field (e.g., LASTNAME + FIRSTNAME). The first field in the expression is the primary sort field, the second is the secondary sort field, and so on.

There are other sorting possibilities but they are beyond the scope of this basic guide.

Activating an Index

To make dBase IV use your index:

1. From the Database Design screen, select the Organize pull-down menu. Highlight the *Order Records by Index* option and press **Enter**.

2. Highlight the desired option from the submenu listing index options and press **Enter**.

3. Press **F2** to view the data in sorted order. If the Edit screen appears instead of the Browse screen, press **F2** again to switch. If necessary, press **PgUp** to scroll to the first record.

⬚S⬚ earching a Database

dBase IV can search a database for a specific record or grouping of records with something in common.

Searching for Specific Records

Both the Browse screen and Edit screen menu bars include the Go To menu, which provides options for basic searches. The first four options on the Go To menu let you position the Edit or Browse screen highlight on a particular record, based on the record's position in the database file.

Forward and Backward Searches

The bottom half of the Go To menu on the Edit and Browse screen menu bars displays the *Forward Search* and *Backward Search* options.

1. Bring up the Browse or Edit screen, go to the position in the database you want to search from (e.g., select Top Record from the Go To menu to go to the beginning of the database), and move the highlight to the field you want to search.

2. Call up the Go To menu and select *Forward (or Backward) Search.*

3. When dBase displays the prompt *Enter search string,* type the text/numbers you want to search for and press **Enter**. To locate items that match a pattern rather than an exact value, you can use the wildcard character **?** to match a single character or the wildcard character ***** to match any group of characters.

Index Searches

The index search searches the database index rather than the database file and usually finds information in large databases more quickly than a forward or backward search. The are two stages of an index search: First, using the Database Design screen, you activate the index that contains the field you want to search. Second, using the Browse or Edit screen, you perform the search.

1. Highlight the database name in the Data panel of the Control Center and press **Shift-F2 (Design)**.

2. From the Organize pull-down menu, select *Order Records by Index.*

3. From the submenu, select the desired index. The field you search must be either the sole field indexed on, or the first field in the index expression.

4. Press **F2 (Data)** to go to the Browse or Edit screen to conduct your search.

5. Select the *Index Key Search* option from the Go To menu. dBase presents the prompt *Enter search string for <field name>.* This is the expression for the current index; dBase displays this to tell you what field you currently can search.

6. Type the value to search on (including upper- and lower-case letters) and press **Enter**. dBase will locate and display the first record that matches your search string.

Q uerying the Database

The process of selecting specific records from the database is called querying the database. The technique you use is called *query by example*, abbreviated QBE. dBase IV presents a skeleton of the database file in use, and you give examples of the kinds of information you want dBase to display. Queries are handled via the Query Design screen. Keys used to design a database query are described below:

KEY   **EFFECT**

Enter   Completes an entry in the file or view skeleton

Arrow-Left,   Moves the cursor one character to the left or right in the file or
   -Right   view skeleton

Arrow-Up,   Moves highlight up or down one row in the file skeleton
   -Down

PgUp,PgDn   Moves to the next or previous page of file skeletons

Ctrl-PgUp,   Moves to the top or bottom of the file skeleton column
   -PgDn

Home   Moves cursor to the first field in the file or view skeleton

End   Moves cursor to the last field in the file or view skeleton

Ctrl-End   Saves changes and exits the Query Design screen

Tab   Moves the cursor one field to the right in the file or view skeleton

Shift-Tab   Moves the cursor one field to the left in the file or view skeleton

Backspace   Moves cursor one space back, erasing along the way, in the file or view skeleton

Del   Deletes the character the cursor is on, in the file or view skeleton

F3 (Prev)   Moves back to the previous skeleton

F4 (Next)   Moves forward to the next skeleton

F5 (Field)   Deletes or reinserts all fields in the view skeleton

F6 (Select)   Highlights a group of adjacent fields in the view skeleton

F7 (Move)   Moves selected text to a new location in the view skeleton

F9 (Zoom)   Expands and shrinks skeleton columns and condition boxes

Shift-F2 (Design)   Brings up a Query Design screen for the selected database

Ctrl-N   Inserts a blank row between two existing rows in the file skeleton

Ctrl-U   Deletes the entire current row in the file skeleton or condition box

Ctrl-W   Saves changes and exits the Query Design screen

Ctrl-Y   Deletes the entire current row in the view skeleton

Esc   Exits without saving changes and returns to the previously used screen

The Query Design Screen

Before you can query a database, you must make sure that the database file is open, and then you must get to the Query Design screen.

1. Press **Shift-F2 (Design)** from the Edit or Browse screen, or use the arrow keys to highlight <create> in the Queries panel from the Control Center and press **Enter**. (After you've viewed or worked with the results of a query, you can return to the Query Design screen by selecting *Transfer to Query Design* from the Exit pull–down menu.) In addition to the usual menu bar at the top of the screen and the status bar and navigation line at the bottom, the Query Design screen includes a **file skeleton** and a **view skeleton**.

The **file skeleton**, near the top of the screen, displays the names of all fields in the database; you will use it to specify search criteria. In the leftmost column is the name of the database file that the skeleton represents. Each field name in the database is listed in boxes to the right of the database name. You can use the **Home, End, Tab, and Shift-Tab** keys to scroll left and right through these field names and to view those that are off the edge of the screen. (Although the Query Design screen initially contains only one file skeleton, you can add as many as 7 more, enabling you to construct complex queries involving several databases.)

2. To specify the records you want a query to display, you enter "filter conditions" in the file skeleton. You put the filter condition under the appropriate box (highlight the field and enter the filter condition in the field box) and then press **F2 (Data)** to see the results. If the Edit screen appears, press **F2** again to switch to the Browse screen. (You can remove filter conditions using the file skeleton's navigation and editing keys.)

Saving a Query for Future Use

1. At the Query Design screen, after creating and testing the query using **F2**, highlight the _Layout_ option in the menu bar and select _Edit Description of Query._ When prompted, type a description and press **Enter**.

2. Select _Save This Query_ from the Layout menu or _Save Changes and Exit_ from the Exit menu. When dBase displays the prompt _Save as:_, enter a valid DOS file name and press **Enter**. dBase will automatically add the extension .QBE to the file name you provide.

Activating a Saved Query

1. At the Control Center, highlight the name of the query in the Queries panel and press **F2**.

2. Three options will be displayed: _Use View, Modify Query,_ and _Display Data._ Selecting **Display Data** activates the view and takes you to the Edit or Browse screen. Selecting **Use View** puts the view name above the line in the Queries panel.

3. When finished, select _Exit_ from the Exit pull–down menu to return to the Control Center. The view name will be above the line in the Queries panel, indicating that it is still active.

Changing a Query

1. Highlight its name in the Queries panel and press **Shift-F2 (Design)**.

2. Modify the query using the same techniques used to create queries. You can see the results by pressing **F2 (Data)**. Return to the Query Design screen by pressing **Shift-F2 (Design)**.

3. When finished, select _Save Changes and Exit_ from the Exit pull–down menu to return to the Control Center. The query name will still be accessible.

Deactivating a Query

You must deactivate a query to remove its effect on your view of the associated database. Simply opening the database file deactivates the query. Highlight the database name in the Data panel and press **F2 (Data)**.

R eports

Creating a Custom Report

Only basic procedures for creating a customized report are included here. There are many more features of dBase IV not covered that you may want to look into, to create a report that fits your specific needs.

1. Ensure that the database is currently in use and, from the Control Center, highlight <create> in the Reports panel and press **Enter**. The Reports Design screen appears. The ruler just beneath the menu bar on the Reports Design screen shows margins and tab stops. The center of the screen is divided into five bands. Each band corresponds to a section of the printed report. Anything that you place within a band is printed only in the corresponding section of the report. The five bands are as follows:
   a. Page header: printed once at the top of each page.
   b. Report intro: printed once at the beginning of the report.
   c. Detail: the body of the report. Typically, this section displays records from a database file.
   d. Report summary: printed once at the end of the report and can be used to display totals or closing information about the report.
   e. Page footer: printed once at the bottom of each page and can be used to display page numbers or other useful information.

2. Select *Quick Layouts* from the Layout menu. Select a layout from among the three options provided:
   a. Column Layout: columnar listings similar to the quick report.
   b. Form Layout: forms using a format similar to the Edit screen for records.
   c. Mailmerge Layout: form letters with information from a database inserted into predetermined locations in the letter.

3. After selecting a general format for your report, select the margins. When printing on 8.5 x 11-inch paper, a right margin setting of about 64 will give generally adequate left and right margins on the printed page.
   a. Pull down the Words menu from the menu bar and select *Modify Ruler*.
   b. Press **Tab** or **Arrow-Right** to move to the left-margin column, then type "[" to mark the left-margin position. Then use **Arrow-Right** or **Tab** to position the cursor to the right-column margin, and type "]" to mark the right margin.

c. Press **Ctrl-End** to finish modifying the ruler and record changes.

4. Edit the field templates and text to your satisfaction.
   a. To edit a band, move the cursor to the band border(which contains the band name). If the band is closed (no blank space beneath the border), you need to open it by pressing **Enter** and moving the cursor into the band.
   b. Now you can enter and edit text, move fields around, and modify the contents of the band using **F5** to place fields in the band. You can close a band that you are not using by moving the cursor to the band border and pressing **Enter**. Just remember to reopen the band before printing because closed bands do not print.

5. A report band can be modified using either the layout editor or the word–wrap editor. Only the layout editor will be discussed here. The layout editor lets you place fields on the band, move them around, and add fields, boxes, and lines to the report.
   a. Adding fields: Position the cursor where you want the leftmost character of the field template to appear. Either press **F5 (Field)** or select *Add Field* from the Fields pull–down menu. Select a field, by positioning the highlight and pressing **Enter**, from the submenu of possible fields:
      1) <u>database</u>: those fields available from the current database or query
      2) <u>calculated</u>: display the results of dBase calculations
      3) <u>predefined</u>: those that dBase provides and handles automatically, including date, time, recno (record number), and pageno (page number)
      4) <u>summary</u>: average, count, max, min, sum, std, and var
   Press **Ctrl-End** to place the field and return to the design screen. A series of Xs (or Hs or Vs) fills in a portion of the screen. These are the field template, which show the max space that data will occupy on a printed page.
   b. Moving and copying fields: Place the cursor in a corner of the area you want to move and press **F6 (Select)**. If you are moving only a field template, press **Enter**. If you are moving more than a single template, use the arrow keys to highlight the entire area that you want to move or copy and then press **Enter**. Press either **F7 (Move)** or **F8 (Copy)**. Move the highlight to the new location and press **Enter**.
   c. Deleting fields: Move the cursor to the field template and either press **Del** or select *Remove Field* from the Fields menu. To delete an entire line, including all fields and text, move the cursor to the appropriate line and press **Ctrl-Y**.
   d. Adding and deleting text: To <u>insert</u> text into an existing format, position the cursor where you want the new text to appear. Make sure that Insert mode is on and then type your text. To <u>change</u> existing text, position the cursor on the text to be changed and activate Overwrite mode (press **Ins** until the Ins indicator disappears); when you type your changes, the new characters will replace existing characters. To <u>delete</u> text from the format, position the cursor on the character you want to delete and press **Del**. You can delete an entire section of text and fields by using the **Select** key (**F6**) to highlight the area you want to

delete; after pressing **Enter** to complete the selection, press **Del** to delete the entire highlighted block.

e. Adding and deleting lines: To <u>insert</u> a new line on a report format, position the cursor where you want the new line to appear and then press **Ctrl-N**, select *Add Line* from the Words pull-down menu, or press **Enter** while in Insert mode. To <u>delete</u> a line from a report format, move the cursor to the line and either press **Ctrl-Y** or select *Remove Line* from the Words menu.

The keys for editing report formats are:

<u>KEY</u>	<u>EFFECT</u>
Enter	If Insert mode is off, moves down one row; if Insert mode is on, inserts a new line
Arrow-Left, -Right	Moves left or right one character or to end of field template
Arrow-Up, -Down	Moves up or down one row
PgUp,PgDn	Moves to top or bottom of screen
Home	Moves to beginning of line
End	Moves to end of line
Tab	Moves to next tab setting
Shift-Tab	Moves to previous tab setting
Ins	Toggles Insert mode on/off
Backspace	Erases character to the left
Del	Deletes character, field template, or block selected with **F6**
F1 (Help)	Provides help
F5 (Field)	Adds a new field template or changes currently highlighted one
F6 (Select)	Selects field template or block
F7 (Move)	Moves field or block selected with **F6**

111

F8 (Copy)   Copies field or block selected with **F6**

Shift-F7   Changes size of currently selected field template
(Size)

Ctrl-N   Inserts a new line

Ctrl-T   Removes word or field to right

Ctrl-Y   Removes entire line

Esc   Abandons current format without saving changes

Saving Report Formats

1. Highlight Layout on the menu bar and select *Edit Description of Report*. Type a description and press **Enter**.

2. Pull down either the Layout or Exit menu and select *Save This Report* or *Save Changes and Exit*, respectively. When dBase presents the prompt *Save as:*, type a valid DOS file name and press **Enter**. After dBase has written a program for itself to print the report later, it will return you to the Control Center where you'll see the new report name in the Reports panel.

Modifying Report Formats

1. Highlight the name of the format in the Reports panel of the Control Center and press **Shift-F2 (Design)**. You can also press **Enter** and then select *Modify Layout* from the submenu that appears.

2. You'll be put in the report format design screen where you can make the changes you desire.

3. After you've modified your format, save your work by selecting *Save Changes and Exit* from the Exit menu. If you do **not** want to save your changes, select *Abandon Changes and Exit* instead. You'll be returned to the Control Center.

4. If you want to save your modified format as a new file (leaving the original, unmodified format intact), select *Save This Report* from the Layout menu and provide a new file name. Then select *Abandon Changes and Exit* from the Exit pull-down menu.

P rinting

The following sections will guide you through printing the database structure, printing a quick report, and printing customized reports.

Printing the Database Structure

You can print your database structure fairly easily from the Database Design screen.

1. Highlight the database file name in the data panel and press the **Shift-F2** key (Design).

2. Select *Print Database Structure* from the Layout pull–down menu.

3. Select *Begin printing.*

4. Select *Abandon Changes and Exit* from the Exit menu to return to the Control Center.

Printing a Quick Report

dBase IV has a quick method of printing out a report of a file's records.

1. Move the pointer to the database file name in the Data panel or the view in the Queries panel which you want to print. Press **Shift-F9**. The Print box appears.

2. With the *Begin printing* option highlighted, press the **Enter** key. Your print job is sent to the network printer.

Printing Customized Reports

1. Make sure the report format name is highlighted in the appropriate panel of the Control Center and press **Enter**.

2. From the options that appear, select *Print Report.*

3. From the Print menu that appears next, select either *View Report on Screen* (to see a screen display only) or *Begin Printing* (to actually print the text). If you select *Begin Printing*, your job is sent to the network printer immediately.

1. Return to the Control Center.

2. Press **F10** and move to the Exit menu.

3. Move the pointer to *Quit to DOS* and press the **Enter** key. You should be returned to the DOS prompt in the 1DIR menu of the network.

4. If you accidentally select *Quit to Dot Prompt,* type "QUIT" at the dot prompt and press **Enter.**

Bibliography

Cowart, Robert, <u>The ABC's of dBASE IV 1.1</u>, Sybex, Inc., 1991.

Simpson, Alan, <u>dBASE IV Programmer's Instant Reference</u>, Sybex, Inc., 1989.

Simpson, Alan, <u>dBASE IV User's Instant Reference</u>, Sybex, Inc., 1989.

Simpson, Alan, <u>Understanding dBASE IV</u>, Sybex, Inc., 1989.

Index

LIST OF REFERENCES

Adams, Kay A., Halasz, Ida M., and Adams, R. Jerry, Handbook for Developing Computer User Manuals, D.C. Heath and Company, 1986.

Axelrod, C. Warren, Computer Productivity, John Wiley & Sons, Inc., 1982.

Bailey, Robert W., Human Errors in Computer Systems, Prentice–Hall, 1983.

Beard, Richard E. and Callamars, Peter V., A Method for Designing Computer Support Documentation, AFIT/LSH, September 1983.

Borland, Russell E., "'Those Silly Bastards': A Report on Some Users' Views of Documentation", ACM SIGDOC Conference Proceedings Second International Conference on Systems Documentation, The Association for Computing Machinery, Inc., 1984: 11–15.

Bradford, Annette N., "Enhanced User Interface Through Computer Tutorials", IEEE Professional Communication Society Conference Record, IEEE Service Center, 1983: 131.

Brockmann, R. John, "Where Has the Template Tradition in Computer Documentation Led Us?", ACM SIGDOC Conference Proceedings Second International Conference on Systems Documentation, The Association for Computing Machinery, Inc., 1984: 16–18.

Brockmann, R. John, Writing Better Computer User Documentation, John Wiley and Sons, Inc., 1990.

Bryant, Tom, "Good Operator Documentation is Good Business", Technical Support, February 1990: 79–82.

Carroll, John M. and Rosson, Mary Beth, Interfacing Thought: Cognitive Aspects of Human–Computer Interaction, MIT Press, 1987: 83–111.

Chinell, David F., System Documentation: The In–Line Approach, John Wiley and Sons, Inc., 1990: 7–78.

Chorafas, Dimitris N., The Software Handbook, Petrocelli Books, Inc., 1984.

IEEE Standard for Software User Documentation, ANSI/IEEE Std 1063–1987, The Institute of Electrical and Electronics Engineers, Inc., 1988.

Costa, Linda C. and Glasgow, Mary R., "Good Software Demands Good Documentation", IEEE Professional Communication Society Conference Record, IEEE Service Center, 1986.

Derfler, Frank, J. Jr., "An Overview of Networking Options", PC Magazine, December 9, 1986: 141.

Etz, D., "A Standard Development Process for User Publications?", ACM SIGDOC Conference Proceedings Second International Conference on Systems Documentation, The Association for Computing Machinery, Inc., 1984.

Gleason, Jim, "Manual Dexterity – What Makes Instructional Manuals Usable?" IEEE Professional Communication Society Conference Record, IEEE Service Center, 1986.

Glossbrenner, Alfred, How to Look it up Online, St. Martin's Press, 1987.

Grimm, Susan J., How to Write Computer Documentation for Users, Van Nostrand Reinhold Company, 1987.

Grice, Roger A., Text, Context, and Hypertext, MIT Press, 1988: 133–148.

Houghton–Alico, Doann, Creating Computer Software User Guides from Manuals to Menus, McGraw–Hill, Inc., 1985.

Johnson, Bob, "User–Centeredness, Situatedness, and Designing the Media of Computer Documentation", ACM, 1990: 55–60.

Krull, Robert and Rubens, Phillip, "Online Information Content and Navigation", IEEE Professional Communication Society Conference Record, IEEE Service Center, 1986.

Luhn, Robert, "The Organization LAN", PC Magazine, February, 1985: 72.

Maynard, John, "A User–Driven Approach to Better User Manuals", IEEE transactions on Professional Communication, March, 1982: 216–219.

McGrath, Richard A., Where's the Manual? Preparing and Producing the Software User's Manual, Van Nostrand Reinhold Company, 1991.

McNamara, John E., Local Area Networks, Digital Press, 1985.

Pesante, Linda Hutz, "Integrating Writing into Computer Science Courses", SIGCSE Bulletin, Volume 23, Number 1, March 1991: 205-209.

Price, Jonathan, How to Write a Computer Manual, The Benjamin/Cummings Publishing Company, 1984.

Rosenbaum, Stephanie and Walters, R. Dennis, "Audience Diversity: A Major Challenge in Computer Documentation", IEEE Professional Communication Society Conference Record, IEEE Service Center, 1986.

Ruben, Martin L., Documentation Standards and Procedures for Online Systems, Van Nostrand Reinhold Company, 1979.

Rubens, Brenda K., "A Comparison of Hardcopy and Online Tutorials", IEEE Professional Communication Society Conference Record, IEEE Service Center, 1986.

Sachs, Jonathan, "Six Leading LANs", PC World, February, 1985: 108.

Schneiderman, Ben, Designing the User Interface: Strategies for Effective Human-Computer Interaction, Addison-Wesley Publishing Co., 1986: 358-390.

Stimely, Gwen L., "A Stepwise Approach to Developing Software Documentation", ACM, 1990: 122-124.

Sullivan, Patricia A. and Porter, James E., "How Do Writers View Usability Information? A Case Study of a Developing Documentation Writer", ACM, 1990: 29-35.

Weiss, Edmond H., How to Write a Usable User Manual, ISI Press, 1985.

Williams, Patricia A. and Beason, Pamela S., Writing Effective Software Documentation, Scott, Foresman and Company, 1990.

Wilton, Judith A., "What PC Owners Say About Documentation", Proceedings of the International Technical Communications Conference, Society for Technical Communication, 1985: RET-10 - RET-12.

Zirinsky, Mark, "Usability Testing of Documentation", IEEE Professional Communication Society Conference Record, IEEE Service Center, 1986: 25.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center   2
   Cameron Station
   Alexandria, Virginia 22304–6145

2. Superintendent   2
   Attn: Library, Code 0052
   Naval Postgraduate School
   Monterey, California 93943–5000

3. Commanding Officer   3
   Attn: LCDR Nancy K. Clark
   FOSIF WESTPAC BOX 271
   PSC 478H
   FPO AP 96313–1801

4. Professor Norman F. Schneidewind , Code AS/Ss   1
   Administrative Sciences Department
   Naval Postgraduate School
   Monterey, California 93943–5000

5. Professor Myung W. Suh, Code AS/Su   1
   Administrative Sciences Department
   Naval Postgraduate School
   Monterey, California 93943–5000